



# ToPlan

Make It Easy  
Make any Plan

Ignacio Pedrós Gil  
Rafael Olaya Lopez  
Adrián Pérez González  
Jorge Miguel Cervera Gomez  
Adrián Javier Ballesteros Romero



**ToPlan**



<b>Introducción</b>	<b>5</b>
<b>Estudio de viabilidad económica de ToPlan</b>	<b>6</b>
Datos personales del equipo	6
Estudio de mercado	6
Análisis económico, técnico, humano y de viabilidad	8
Análisis DAFO	11
Conclusión de la viabilidad del proyecto	13
<b>POLÍTICA DE PRIVACIDAD</b>	<b>14</b>
¿QUÉ SON LOS DATOS PERSONALES?	14
PRINCIPIOS PARA EL TRATAMIENTOS DE DATOS	15
FINALIDAD	15
MAYORES DE EDAD	16
LEGITIMACIÓN	16
CATEGORÍA DE LOS DATOS RECABADOS	16
CONSENTIMIENTO AL TRATAMIENTO	18
EXACTITUD Y VERACIDAD DE LOS DATOS	18
CUMPLIMIENTO DE LA NORMATIVA DE APLICACIÓN	18
MEDIDAS DE SEGURIDAD	19
EJERCICIO DE DERECHOS	19
DESTINATARIOS Y ENCARGADOS DEL TRATAMIENTO	20
¿NO DESEA RECIBIR INFORMACIÓN DE NOSOTROS O DESEA REVOCAR SU CONSENTIMIENTO?	21
ACEPTACIÓN, CONSENTIMIENTO Y REVOCABILIDAD	21
<b>Documentación APP:</b>	<b>29</b>
Inicio:	30
Registro:	31
Login:	32
Explore:	34
Detalles:	35
<b>Documentación BackOffice:</b>	<b>35</b>
Login	35
Users	36
Esta página se encarga de mostrar todas las acciones relacionadas con el manejo de usuarios	36
Events	43
TypePlan	48





---

<b>Documentación API/BD</b>	<b>51</b>
BASE DE DATOS:	51
WEB API	52
Users	52
Events	58
Typeplan	71





---

## Introducción

Proyecto final del ciclo de FP Superior en Desarrollo de Aplicaciones Multiplataforma hecho por Adrián Javier Ballesteros Romero, Adrián Pérez González, Nacho Pedrós Gil, Rafa Olaya López y Jorge Miguel Cervera Gómez.















Nuestro proyecto tiene como idea la creación de una aplicación móvil para crear y participar en diferentes eventos, una Web Api que recupere e introduzca datos en una base de datos y por último un backOffice para administrar la base de datos.



## Documentos SGE

### Estudio de viabilidad económica de ToPlan

#### 1. Datos personales del equipo

EMPLOYEES								
	Idemployee_1	Name	Email	Surname	Telephone	Nif	Address	Dischargedate
  	2	Jordi	jocego@floridauniversitaria.com	Cervera Gomez	686652681	56706894J	Calle Alqueria Nova	01/02/2021
  	3	Adrian	adriperezgfb@gmail.com	Perez Gonzalez	634809734	20056704Z	C/ Beniarjo 6	01/02/2021
  	4	Nacho	nacho.pedrosgil@outlook.com	Pedros Gil	674977785	19450368C	Avenida san Lorenzo 69	01/02/2021
  	5	Adrian Javier	samilbiza@gmail.com	Ballesteros Romero	682604065	45689752W	C/ Asturias n1	01/02/2021
  	6	Rafa	raolio@floridauniversitaria.es	Olaya Lopez	670896324	24986326Y	C/ Jaime primero	01/02/2021

#### 2. Estudio de mercado

# fever

Es una aplicación gratuita que te permite elegir entre una gran variedad de eventos. Esta aplicación te pide crearte una cuenta, o entrar con email o facebook, luego te pide que elijas la ciudad en la que vives y tus gustos o preferencias. Con estos datos la aplicación filtra los eventos cercanos a ti y te propone los que más se ajustan a tus gustos. También tiene una opción de buscador que te permite filtrar manualmente todos los tipos de eventos disponibles. También tiene la opción de chat para hablar con tus amigos o conocer nuevas personas. La última cosa a destacar sería que esta aplicación tiene la opción de manejar compras, como entradas, dependiendo del tipo de evento.

**Ventajas:** La aplicación es muy sencilla de usar y en segundos puedes configurarla para empezar a usarla con todo su potencial. Tiene muchos tipos de eventos y es muy visual. Los filtros disponibles en el buscador son bastante buenos y la opción de hacer las compras directamente desde la aplicación es algo muy cómodo para el usuario.

**Desventajas:** La aplicación usa ciudades para buscar los eventos, lo que hace que si vives en un pueblo que no sea muy grande o conocido probablemente tengas que poner



**ToPlan**



la ciudad más cercana. A pesar que el tema de hacer las compras desde la aplicación puede ser bastante cómodo, acapara mucho la atención, por lo cual una persona que quiera buscar eventos gratuitos probablemente deseche la aplicación al poco tiempo.



Es una aplicación para buscar planes que hacer en tu ciudad de todo tipo en el que puedes reservar plazas en los eventos que hay disponibles o crear los tuyos propios, tiene un sistema similar a instagram para poder dar like a los eventos, puedes ver comentarios, participantes y el organizador del evento con pulsar sobre él.

**Ventajas:** es sencilla de usar, se organiza por categorías y puedes ver los likes que tienen los planes

**Desventajas:** como principal contra es que la aplicación solo permite hacer planes en las provincias de valladolid y zaragoza cosa que en nuestra aplicación solventaremos permitiendo en todo el territorio nacional, no tiene un servicio de filtro con lo que se puede subir cualquier tipo de plan siendo un poco molesto encontrar anuncios que no son de planes.



Es una plataforma en la que los usuarios crean grupos en base a un interés, como por ejemplo, la playa, la montaña, etc. La gente se une a estos grupos principalmente para reuniones físicas o virtuales. El tema que escogen ellos como "planes" es uno de los aspectos más interesantes de Meetup, porque la mayoría de estos grupos no van más allá de simples "foros" y te dan la posibilidad real de conocer gente. Además, a la hora de crear un grupo, las personas que accedan a él, se comprometen de tal forma, que forman parte de las actividades reales que se organizan para que las personas se conozcan y se presenten. Los usuarios de la aplicación pueden crear un grupo en base a tu propio interés o formar parte de uno ya existente que cumpla con tus mismos intereses.



**ToPlan**



**Ventajas:** Es una buena herramienta para conocer a todo tipo de personas de tu mismo interés que estén cerca de ti.

**Desventajas:** La creación de grupos está muy limitada por parte de la aplicación, ya que no recoge una gran variedad de grupos de interés para que entren los nuevos usuarios. Además, al ser obligatorio pagar por la creación de un grupo para hacer planes, los usuarios se cansarán y no utilizarán la aplicación porque no encontrarán planes de su interés.

### 3. Análisis económico, técnico, humano y de viabilidad

#### Coste de la aplicación y mantenimiento

##### Proyecto ToPlan

Inicio	1-feb
Fin	15-feb
Días efectivos	14

	Coste x día	Coste Total
<b>Desarrollo App</b>	<b>359,36 €</b>	<b>5.031,00 €</b>
Software	- €	- €
Infraestructura AWS - Máquina virtual para API	16,50 €	231,00 €
Desarrollo APP	342,86 €	4.800,00 €
	Coste x día	Coste Total
<b>Mantenimiento Anual</b>	<b>16,50 €</b>	<b>15.622,50</b>
Software	- €	
AWS - Máquina virtual para API	16,50 €	6.022,50 €
Mantenimiento APP	--	9.600,00 €



**ToPlan**





<b>Desarrollo Proyecto</b>			
Inicio	1-feb		
Fin	15-feb		
Días efectivos	14		
FTE	5	perfil desarrollador en formación	
Horas x día x FTE	8	tiempo efectivo	
Horas totales	560		
<b>Economics</b>			
Coste x hora	8,57 €	coste desarrollador en formación	
<b>Coste total</b>	<b>4.800,00 €</b>		
<b>Mantenimiento anual</b>			
Horas totales	480	perfil desarrollador junior	
<b>Economics</b>			
Coste x hora	20,00 €		
<b>Coste total</b>	<b>9.600,00 €</b>		
<b>Lanzamiento</b>			

Dedicación			
Tareas	%	Horas, #	Coste
Desarrollo Front-End	36%	201,6	1.728,00
Desarrollo Back-End	48%	268,8	2.304,00
Diseño Gráfico	6%	33,6	288,00
Gestión proyecto	4%	22,4	192,00
Compliance & economics	6%	33,6	288,00
	<b>100%</b>	<b>560,0</b>	<b>4.800,00</b>



**ToPlan**



## Ganancias mínimas a cumplir una vez puesta en marcha la app

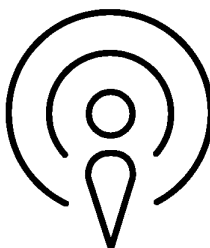
Usuarios diarios: 1500

Clics por usuario Adsense 25%

Precio por click 0.08€

	Ganancias	Clics
Diario	30,00 €	375,00 €
Mensual	900,00 €	11.250,00 €
Anual	10.950,00 €	136.875,00 €

### 4. Análisis DAFO



# ToPlan

### INTERPRETACIÓN

ToPlan una aplicación móvil en la que los usuarios organizan quedadas grupales para realizar todo tipo de actividades de ocio, como deporte, gastronomía o entretenimiento en general.



**ToPlan**



### DEBILIDADES

- Necesidad de volumen de usuarios elevado para hacer atractiva la aplicación y poder contar con diversidad de planes.
- Ausencia de un plan de comercialización y posicionamiento ASO.
- Desconfianza en los usuarios a la hora de relacionarse con desconocidos.
- La necesidad de facilitar información personal para utilizar la aplicación (nombre, apellidos y fecha de nacimiento) puede generar resistencia en los usuarios.
- Carencia de funcionalidad web.
- La idea puede ser fácilmente imitada.

### FORTALEZAS

- Aplicación gratuita.
- Diversidad de oferta con actualización constante por los usuarios: buenos restaurantes, lugares que visitar... tanto nacional como extranjero.
- Inmediatez de los planes, no hay tiempo de espera a la hora de conectar con otros usuarios para apuntarse a los planes.
- Biblioteca de planes accesible a todos los usuarios con sugerencias.
- Aplicación social, los usuarios conocen a otros usuarios sin sentirse amenazados por estar en un entorno grupal.
- Los planes se pueden compartir fuera de la aplicación.
- Bajos costos operacionales.

### AMENAZAS

- Modelo de negocio impredecible por el entorno COVID-19. Las restricciones COVID-19 y peligro de contagio, pueden penalizar el objeto de la aplicación.
- Incompatibilidades entre los usuarios que se juntan en un plan que puede llevar a malas críticas en el App Store (ideológicas, afinidades, edad...).
- Aplicaciones similares en el mercado (Wikiloc, Click and Plan, Meetup)



**ToPlan**



- 
- No encontrar partners económicos que hagan viable económicamente la aplicación. (anunciantes...)

### OPORTUNIDADES

- Ingresos adicionales en modelo b2b por acuerdos con partners afines (restaurantes, museos, gimnasios...)
- Posibilidad de realizar acuerdos con partners para disponer de ventajas exclusivas en un plan (descuento en restaurantes, guía para una ruta de trekking...).

## 5. Conclusión de la viabilidad del proyecto

Como principal contra es la carencia de usuarios en la puesta en marcha de la aplicación, con un estudio de marketing y comercialización empezaríamos seguros de poder contar con la base suficiente de usuarios para que la aplicación se mantuviese sin pérdidas.

Sabemos que esto no pasará hasta mínimo el primer año, con una base mínima de 1500 usuarios diarios la aplicación sería capaz de mantenerse con los gastos que genera el mantenimiento anual. Pasado ese año ya estaría establecida en el mercado y sumando las promociones de los restaurantes y los eventos con partners las ganancias reales serían notorias a partir del primer año.

Una monetización férrea con la capacidad de escalar a diferentes países, e posicionamiento en la Play Store será clave para el éxito de nuestro proyecto.

Agradezco que estés en esta página, eso significa que tus datos te importan, y quieres conocer el destino de los mismos y quién los recoge.



**ToPlan**



## POLÍTICA DE PRIVACIDAD

De acuerdo con el artículo 10 de la Ley 34/2002, de Servicios de la Sociedad de la Información y Comercio Electrónico, ponemos en su conocimiento la siguiente información:

- El responsable de tratamiento es ToPlan que es de titularidad de Florida Centre de Formació, S. C. V., (en adelante FLORIDA), domiciliada en C/ Rey Jaime I, nº 2, 46470 Catarroja, Valencia. FLORIDA, con C.I.F. F-46278669, está inscrita en el Registro de Cooperativas de la Comunidad Valenciana, folio CV-1, asiento nº 36, con fecha 30.07.1996.
- Email: "[toplan@outlook.es](mailto:toplan@outlook.es)"
- La actividad social es: Proyecto integrado final de ciclo superior cursando segundo curso de Desarrollo de Aplicaciones Multiplataforma con el tutor de la promoción Manel Viel Córdova.

La presente Política de Privacidad será válida únicamente para los datos de carácter personal obtenidos en la App "ToPlan", siendo de aplicación para las páginas online propiedad de la empresa, como son [www.floridauniversitaria.es](http://www.floridauniversitaria.es) no siendo aplicable para aquella información recabada por terceros en otras páginas web, incluso si éstas se encuentran enlazadas por el Sitio Web.

Con ello, se manifiesta por parte del responsable el compromiso por mantener y garantizar las relaciones comerciales de forma segura mediante la protección de los datos personales y velando por el cumplimiento del derecho a la privacidad de cada uno de los usuarios del sitio web, reflejando que no es necesario evaluación de impacto ni delegado de protección de datos.

Las presentes políticas son revisadas con cada cambio significativo y como mínimo una vez al año.

## ¿QUÉ SON LOS DATOS PERSONALES?

Una pequeña aproximación es importante, por ello, el USUARIO debe saber que un dato de carácter personal sería cualquier información relativa a una persona que nos facilita cuando visita nuestro sitio web, en nuestro caso nombre y apellidos además del correo electrónico asociado a tu cuenta una vez registrada en la aplicación.



**ToPlan**



## PRINCIPIOS PARA EL TRATAMIENTOS DE DATOS

Para tratar sus datos personales, se aplicará según el Reglamento General sobre Protección de Datos y la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales los siguientes principios:

- **Principio de licitud, lealtad y transparencia:** requerimos su consentimiento para el tratamiento de sus datos personales para uno o varios fines específicos que le informaremos previamente de forma clara y comprensible.
- **Principio de minimización de datos:** Sólo solicitaremos datos estrictamente necesarios en relación con los fines para los que ya se te ha informado previamente al USUARIO.
- **Principio de limitación del plazo de conservación:** como más adelante podrá comprobar, los datos serán mantenidos durante no más tiempo del necesario para los fines del tratamiento, en función a la finalidad, se informará del plazo de conservación correspondiente.
- **Principio de integridad y confidencialidad:** Sus datos serán tratados de tal manera que se garantice una seguridad adecuada de los datos personales y se garantice confidencialidad. Debe saber que tomamos todas las medidas encaminadas a evitar el acceso no autorizado o uso indebido de los datos de los usuarios por parte de terceros.

## FINALIDAD

La finalidad para la que se recabarán los datos que hemos categorizado en el apartado 2.4, serán la siguientes y que posteriormente se detallarán:

Gestionar la lista de suscriptores y usuarios adscritos a la aplicación.

El cumplimiento de las correspondientes obligaciones de índole contable, legal, fiscal y administrativa.

Análítica y publicitaria con el fin de obtener beneficio económico.

Únicamente el titular tendrá acceso a sus datos, y bajo ningún concepto, estos datos serán cedidos, compartidos, transferidos, ni vendidos a ningún tercero.



**ToPlan**



En ningún caso se realizará un uso diferente que la finalidad para los que han sido recabados los datos ni tampoco cederemos a un tercero estos datos.

## MAYORES DE EDAD

En el supuesto de ser mayor de 18 años, podrá registrarse en la aplicación.

¿Qué ocurre en el caso de que seas menor de 18 años?

En este supuesto, será condición obligatoria el consentimiento de sus padres o tutores para que podamos tratar sus datos personales

**Advertencia:** Si tiene menos de 18 años y no ha obtenido el consentimiento de sus padres, no puede registrarse en la aplicación ToPlan por lo que procederemos a denegar su solicitud en caso de tener constancia de ello.

## LEGITIMACIÓN

Gracias al consentimiento, podemos tratar sus datos siendo requisito obligatorio para poder suscribirse a la aplicación ToPlan.

Como bien sabe, puede retirar su consentimiento en el momento que lo desee, entendemos que si retira el consentimiento es a la hora de eliminar su cuenta, puesto que no podrá volver a acceder si no acepta el consentimiento del trato de sus datos.

Estaremos legitimados para tratar sus datos para el cumplimiento de obligaciones legales en materia fiscal, mercantil y contable, para la facturación y llevanza de la contabilidad.

Asimismo, gracias al interés legítimo podemos tratar determinados datos relacionados con la publicidad, mercadotecnia directa y/o el uso de cookies cuando no sea posible a través del consentimiento así como para atender la solicitud de publicar un comentario.



**ToPlan**



---

## CATEGORÍA DE LOS DATOS RECABADOS

Los datos recabados en ningún momento están especialmente protegidos, sino que se encuentran categorizados como datos identificativos.

Los datos de identificación que se deriven del medio de contacto empleado por el cliente, a consecuencia de un servicio, y los que constan en los contratos acordados o los que se derivan de la ejecución de la relación contractual: Nombre y apellidos junto al correo electrónico

– La finalidad de tratar estos datos es para cumplir con las obligaciones y derechos derivados de la relación contractual existente.

### **Datos relativos a Usuarios de la app:**

Los datos de identificación que podemos requerir en el formulario de contacto dispuesto (nombre, apellidos y correo electrónico) y los que nos facilite voluntariamente en el campo de texto establecido.

Los que voluntariamente nos proporcione al ponerse en contacto con nosotros mediante el correo electrónico indicado en la web, que podrán ser de identificación y aquellos otros que voluntariamente se nos faciliten como archivos (documentos, contratos u otros).

La imagen, nombre y apellidos si publicamos una recomendación suya de nuestros servicios, previo su consentimiento.

– Las finalidades para el tratamiento de estos datos están basadas en el interés del titular de los datos y basadas en el interés legítimo del responsable con el objeto de mejorar la experiencia de navegación y analítica de la app.

### **Redes Sociales:**

Los datos identificativos que suministre usted mismo como usuario de la red social.

Los que voluntariamente nos proporcione al ponerse en contacto con nosotros al hacer comentarios o mediante mensajería instantánea de la red.

### **Suscripción a newsletter y/o contenidos:**

Los datos de identificación que requerimos en el formulario de contacto dispuesto para dicha suscripción son (nombre, apellidos y correo electrónico).



**ToPlan**





– La finalidad por la que se recaban estos datos es para ofrecer novedades, descuentos y promociones sobre la base jurídica del consentimiento libre, informado, específico e inequívoco prestado.

#### **Sistemas de rastreo utilizadas en este sitio:**

En la App se utilizan sistemas de rastreo como Google (Analytics), con este tipo de herramientas y en relación con lo anteriormente establecido, también se estudian las preferencias de sus usuarios, sus características demográficas, sus patrones de tráfico, y otra información en conjunto para comprender mejor quiénes constituyen su audiencia y qué es lo que esta necesita. El rastreo de las preferencias de nuestros usuarios también nos ayuda a mostrarle los avisos publicitarios más relevantes.

### **CONSENTIMIENTO AL TRATAMIENTO**

Conservaremos sus datos durante el tiempo legalmente establecido o hasta que solicite eliminarlos.

### **EXACTITUD Y VERACIDAD DE LOS DATOS**

Obviamente el Usuario es el único responsable de la veracidad y exactitud de los datos que me remita eximiendo al equipo ToPlan de Florida de cualquier tipo de responsabilidad al respecto.

Como usuario, debe garantizar la exactitud y autenticidad de los datos personales facilitados debiendo aportar la información completa y correcta en los distintos formularios de captación de datos.

### **CUMPLIMIENTO DE LA NORMATIVA DE APLICACIÓN**

Actualmente nos regimos por el Reglamento Europeo Sobre Protección de Datos así como la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales, siendo por ello entre otras, la normativa que damos cumplimiento.

Asimismo, el responsable de tratamiento informa que da cumplimiento a la Ley 34/2002 de 11 de julio, de Servicios de la Sociedad de la Información y el Comercio Electrónico y le



**ToPlan**



solicitará su consentimiento al USUARIO para el tratamiento de su correo electrónico con fines comerciales en cada momento.

Adicionalmente, el USUARIO consiente el tratamiento de sus datos con la finalidad de informarles, por cualquier medio, incluido el correo electrónico, de productos y servicios relacionados con ToPlan

En caso de no autorizar el tratamiento de sus datos con la finalidad señalada anteriormente, el USUARIO deberá declarar que no quiere recibir información referente a los productos y servicios anteriormente citados.

## **MEDIDAS DE SEGURIDAD**

El responsable de tratamiento le informa que se tienen implantadas las medidas de seguridad de índole técnica y organizativas necesarias para garantizar la seguridad de sus datos de carácter personal y evitar su alteración, pérdida y tratamiento y/o acceso no autorizado, habida cuenta del estado de la tecnología, la naturaleza de los datos almacenados y los riesgos a que están expuestos, ya provengan de la acción humana o del medio físico o natural. Todo ello de conformidad con lo previsto en el actual Reglamento General sobre Protección de Datos y la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales.

Asimismo, se le informa que se han establecido medidas adicionales en orden a reforzar la confidencialidad e integridad de la información en su organización. Manteniendo continuamente la supervisión, control y evaluación de los procesos para asegurar el respeto a la privacidad de los datos.

## **EJERCICIO DE DERECHOS**

Aquellas personas físicas que hayan facilitado sus datos a través del registro en la App, podrán dirigirse al titular de la misma con el fin de ejercitar gratuitamente sus derechos de acceso a sus datos, rectificación o supresión, limitación y oposición respecto de los datos incorporados en sus ficheros, asimismo, podrá presentar una reclamación ante la autoridad de control ante la Agencia Española de Protección de Datos sita en Calle Jorge Juan, 6, 28001, Madrid, <https://www.aepd.es/es> o a través de la [sede electrónica](#).

El método más rápido y sencillo sería acceder a su cuenta de usuario directamente y modificar sus datos o borrar su cuenta de usuario. Cualquier información que necesite



**ToPlan**



almacenar, en virtud de una obligación legal o contractual, será bloqueada y sólo utilizada para dichos fines en lugar de ser borrada.

El interesado, en este caso, usted, podrá ejercitar sus derechos mediante comunicación por escrito dirigida a ToPlan con la referencia “Protección de datos”, especificando sus datos, acreditando su identidad y los motivos de su solicitud en la siguiente dirección:

ToPlan que es de titularidad de Florida Centre de Formació, S. C. V.  
C/ Rey Jaime I, nº 2, 46470 Catarroja, Valencia.

Otra forma de ejercitar los derechos es a través del correo electrónico, escribiendo al siguiente email: [toplan@outlook.es](mailto:toplan@outlook.es)

## DESTINATARIOS Y ENCARGADOS DEL TRATAMIENTO

Los datos no serán comunicados a ningún tercero, salvo obligación legal o que nos sea prestado su consentimiento informado, libre, específico e inequívoco a tal fin. Como encargados de tratamiento ajenos al citado responsable y con carácter meramente informativo debe saber que:

Los servicios de almacenamiento en la nube de los datos de los USUARIOS de ToPlan están en AWS (Amazon Web Services). AWS cumple la norma ISO 27018, un código de conducta diseñado para proteger los datos personales en la nube. Se basa en la norma de seguridad de la información de la ISO 27002 y ofrece directrices de implementación sobre los controles de la ISO 27002 aplicables a la información personal identificable (PII) procesada por proveedores de servicios en la cloud pública. Para obtener más información o ver la certificación ISO 27018 de AWS, consulte la página web relativa a la [conformidad de AWS con la ISO 27018](https://aws.amazon.com/es/compliance/eu-us-privacy-shield-faq/). <https://aws.amazon.com/es/compliance/eu-us-privacy-shield-faq/>

También usamos servicios de almacenamiento en la nube con la compañía Google Inc (Identificada por la marca comercial “Google Drive”) una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94043, Estados Unidos (“Google”). Para más información, puedes consultar su política de privacidad en el siguiente enlace: <https://policies.google.com/privacy?hl=es>

Asimismo, se recogen de forma anónima algunos datos de navegación por Google Analytics, es un servicio analítico de web prestado por Google, Inc., una compañía de Delaware cuya oficina principal está en 1600 Amphitheatre Parkway, Mountain View (California), CA 94043, Estados Unidos (“Google”). Puedes obtener más información en: <https://support.google.com/analytics/answer/9019185>



**ToPlan**



---

### **¿Quiénes son los destinatarios de la información?**

- Agencia Estatal de la Administración Tributaria, así como gestorías a consecuencia de nuestras obligaciones de índole legal, fiscal y administrativa
- Entidades financieras con las que trabajamos para la gestión de cobros

### **¿NO DESEA RECIBIR INFORMACIÓN DE NOSOTROS O DESEA REVOCAR SU CONSENTIMIENTO?**

De conformidad con lo dispuesto en la Ley 34/2002, de 11 de junio de Servicios de la Sociedad de la Información y del Comercio Electrónico puede oponerse al uso de su información para fines publicitarios, investigaciones de mercado o desarrollo de encuestas de satisfacción en cualquier momento, así como revocar su consentimiento en cualquier momento (sin efecto retroactivo).

En relación con la gestión de sus datos asociados a los perfiles sociales de ToPlan, el ejercicio del derecho de acceso, dependerá de la funcionalidad de la red social y las posibilidades de acceso a la información de los perfiles de los USUARIOS. Con relación a los derechos de acceso y rectificación, le recomendamos que sólo podrá satisfacerse en relación a aquella información que se encuentre bajo el control de ToPlan.

Además podrá dejar de interactuar, seguir o recibir información de los perfiles sociales de ToPlan, eliminar los contenidos que dejen de interesarle o restringir con quién comparte sus conexiones, mediante los mecanismos estipulados en las diferentes redes sociales.

### **ACEPTACIÓN, CONSENTIMIENTO Y REVOCABILIDAD**

El Usuario declara haber sido informado de las condiciones sobre protección de datos de carácter personal, aceptando y consintiendo el tratamiento de los mismos por parte de ToPlan en la forma y para las finalidades indicadas en el aviso legal.

Como bien sabe y le hemos comunicado a lo largo de las presentes políticas de privacidad, en cualquier momento podrá revocar sus datos, pero siempre sin carácter retroactivo, siendo consciente que sin el consentimiento de que acepta los términos no podrá acceder a la aplicación.

No obstante, si existe alguna duda no dude en contactar con el equipo vía correo electrónico en la siguiente dirección con el asunto [toplan@outlook.es](mailto:toplan@outlook.es).

ToPlan





---

## DIARIO DIGITAL TOPLAN



### EQUIPO

Adrián Pérez - AdriP  
Rafael Olaya - Rafa  
Ignacio Pedrós - Nacho  
Jorge Miguel Cervera - Jordi  
Adrián Javier Ballesteros - AdriJ

### PREPRODUCCIÓN

**21/12/20:** Todos presentes vía online.

Primera reunión del proyecto, hemos hablado de cómo vamos a desarrollar la aplicación teniendo una ligera idea de que vamos a realizar una aplicación social, viendo el nicho de mercado post entorno COVID-19.

Decisión unánime de nombre de la app ToPlan y el logo diseñado por AdriP.

**28/12/20:** Todos presentes vía online.

Decidimos repartirnos el trabajo, AdriP y Jordi trabajando en el diseño de la aplicación en React-Native y prototipando la aplicación, por otra parte AdriJ, Nacho y Rafa empiezan con la descripción del proyecto a desarrollar.

**05/01/21:** Todos presentes vía online.

Puesta en común del trabajo realizado y preparación de la entrega, decisión de entregarlo solo por un miembro del equipo, subido a Florida Oberta por AdriP.



**ToPlan**



---

## PRODUCCIÓN

**01/02/21:** Todos presentes online.

Diseño de la base de datos, definición de tiempos a la hora de trabajar y repartición de tareas. Hora acordada de la reunión diaria a las 15:15h.

**02/02/21:** Todos presentes presencial menos Jordi online.

Arreglar React-Native pantallas de los componentes y que van en la carpeta de screens.

La API básica está hecha pero falta añadir una opción de Administrador, preparación de encriptación de passwords con una librería de react native para que encaje con el API.

Revisar navegabilidad entre pantallas en React.

HCM parte de SGE que falta por definir y hacer una parte para añadir empleados.

**03/02/21:** Todos presentes vía online.

Backend, faltan acabar los métodos del API y el tema de encriptación de contraseñas para la hora de subir la app se verá el viernes con Belén.

Tema de React-Native aclaramos para unir el proyecto una vez esté acabado.

Página web de administración para la hora de venderla hablamos ayer con Fran.

React-Native y API deben estar finalizados para el viernes 5.

Hablar con Manel para ver el tema de la conexión a la base de datos.

**04/02/21:** Todos presentes de manera presencial.

Cuando esté acabado el tema del API empezaremos con la encriptación de la contraseña así que pasa a segundo plano. El tema de la base de datos vamos a clonar el repositorio en el AWS con una máquina virtual (utilizando la cuenta de AdriP). El tema de la página web del administrador se pondrán Jordi y AdriJ. Creación de dos pantallas en Axure de AdriP que faltaban la de editar perfil, detalles del plan y creación de plan.

**05/02/21:** Todos presentes vía online.

AdriJ la API está hecha, revisando el host online y empezaremos a trabajar en local hasta subirlo lo verán Rafa y AdriJ este fin de semana. Nacho estuvo trabajando sobre la función en base a una fecha de nacimiento que calculase la edad de un usuario y preparando los métodos necesarios.

Hostear la base desde el ordenador y probar conexión desde el otro ordenador y una vez funcione ya estaría lista.

Funcionales de página web por parte de Nacho y Jordi.

Una vez acabadas las pantallas en React empieza la programación de navegación.

**06/02/21:** Todos presentes vía online.

Jordi y Nacho están con la administración de los usuarios en el backoffice y la administración de eventos, en principio hoy estará hecho a falta de las pruebas pertinentes con la base de datos, puesto que no disponen de la base de datos en local (falta hacer



**ToPlan**



testing). En la página de inicio hacer un login pero cuando esté acabado. AdriJ está hosteando el API, en principio hoy debería estar acabado. Rafa estuvo haciendo pruebas del API y hoy se pone con las funcionalidades de react-native las pantallas. AdriP acabando las dos pantallas de React-Native de crear plan y detalles de plan.

**07/02/21:** Todos presentes vía online.

Jordi está implementando con la API menos el delete de users, está pendiente de hablar con AdriJ con el método que si existe un usuario para que te devuelva true, Jordi ha tenido conflictos con el axios y con el desarrollo de la Api. Nacho ha estado acabando la parte de administración de eventos y le queda devolver un evento en específico y insertar eventos. A la espera de hacer algunos métodos que falten con la aplicación.

AdriJ estará con Rafa haciendo funcionalidades y la navegación de React.

Rafa estuvo instalando las librerías pero tuvo problemas con una librería llamada navigation stack. Adri P acabada la pantalla de crear plan y recogiendo información de leyes de protección de datos y política de privacidad.

**08/02/21:** Todos presentes vía presencial y Rafa online por motivos del API en su pc.

Rafa estuvo haciendo ayer las funcionalidades de las pantallas con AdriJ y los métodos del API necesarios para las pantallas, queda pendiente trabajar la navegación con Manel. Por otro lado, queda la creación de la tabla subtipos con la categoría (ocio...) y el subtipo (cine, museo...). AdriP ha terminado las pantallas marcadas para él de react native a falta de hacer componentes, y en la pantalla de detalles del plan ver con AdriJ el Confirmar/Desconfirmar asistencia.

**09/02/21:** Todos presentes vía online.

Jordi ha acabado la página de usuarios a excepción de multiselect y a la espera de filtrar por X usuarios, cuando acabe empezará con una página de inicio que tenga un usuario y contraseña y que solo te deje pasar de esa página a la siguiente si eres administrador. En el usuario hay que comprobar si existe un user y si está hecho eso, comprobará el token para que se guarde en una Cookie, ha comentado Rafa pero hay que esperar a las funcionalidades de React.

AdriJ se encarga del login con el token hoy y cuando lo tenga estará con las Cookies una vez tenga claro lo de login. A Nacho le falta hacer el filtrado por categorías de la pestaña eventos y AdriP está con la instalación del proyecto ToPlan de Flexygo.

La web funcionará a final del miércoles aproximadamente.

React funcionalidades a final del jueves aproximadamente.

**10/02/21:** Todos presentes vía presencial.



**ToPlan**



Jordi, el apartado de users todo realizado y está acabando la pantalla de login del backoffice de la aplicación. Nacho está acabando el filtrado de los eventos y la pantalla del TypePlan (tipos de planes) hoy acabada.

Rafa acabó el registro de usuarios y estuvo arreglando la de “explores” y poniendo los datetimepickers el checkbox de la política de privacidad.

AdriJ estuvo con la autenticación a la espera de correos con Juanmi.

AdriP en principio hoy acabado el HCM añadiendo todos los empleados con la creación de la base de datos y creando el formulario para agregar empleados.

**11/02/21:** Todos presentes vía online.

Jordi dispone del login funcional y está a falta mirar si devuelve las listas visto con Rafa, y pendiente de ayudar con la cookie a AdriJ. AdriJ ayer habló con Juanmi sobre el tema de autenticación de usuarios y ha estado arreglando varios métodos con la documentación y ponerse una vez acabado con las pantallas con la funcionalidad. Rafa acabada la navegación registro y está con la de login. Nacho acabada la página de TypePlan junto al filtrado de eventos le queda arreglar y limpiar el código solucionando una parte de la columna que colapsa con otra además tiene que acabar la página y que se ponga con Jordi y AdriJ.

**12/02/21:** Todos presentes vía presencial.

Rafa tiene hechas las funcionalidades de la app, falta la pantalla de Nacho de react native, faltaría retocar la navegación para que funcione bien todo y la pantalla que hizo AdriP retocarla para adaptarla. AdriJ ayer acabó la pantalla que mando, le faltan retoques visuales que tiene que acabar Rafa y ayudar a Jordi y Nacho aparte de documentar. Jordi documenta y estiliza el backoffice comprobando que todo funciona. Nacho ha acabado el plan de la web y está con la pantalla de detalles. AdriP está trabajando en el inicio del análisis de viabilidad económica, técnica y humana del proyecto.

**13/02/21:** Todos presentes vía online.

AdriJ está de soporte con Nacho, Rafa y Jordi. Poner a funcionar el backoffice con Jordi y preparar la otra máquina virtual para poner a punto todo el API y hacer la estructura de cómo haremos la documentación.

Jordi todas las preferencias a falta de testear el Axios y realizado el merge con Nacho del proyecto.

Nacho acabada lo de detalles de React-Native y modificado con el backoffice.

Rafa todo React-Native y acabada la funcionalidad componente de términos y condiciones en un overlay terminada con la navegación y librería de guardar cookies.

AdriP finalizar el análisis de viabilidad económica, técnica y humana y ayuda con los iconos que faltan de la aplicación.



**ToPlan**





---

**14/02/21:** Todos presentes vía online.

AdriJ acabó documentación de la base de datos y el web API y ahora revisa documentación y pasar a limpio publicando el backoffice, Nacho aplicando el diseño de todas las pantallas del backoffice y redactando documentación técnica. Jordi está preparando el deploy del backoffice para pasarlo a AdriJ y que lo suba a la máquina virtual. Rafa está trabajando en el apk de React-Native y haciendo la build. AdriP acabar el análisis de los ingresos de la empresa y pasar a limpio la documentación del proyecto.

Todo el equipo realizando testeos del apk en nuestros teléfonos.

## **Documentación técnica**

**Github del proyecto:** <https://github.com/Florida2DAM/ToPlan>

**Mockup y logos:** <https://github.com/Florida2DAM/ToPlan/tree/MockUp>

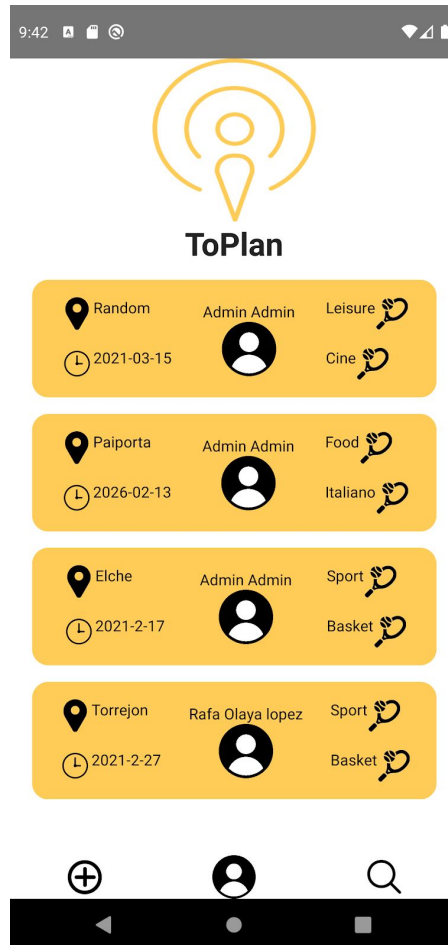


**ToPlan**



## Documentación APP:

Rama de GitHub: [\\_Funcionalidades-Axios-Navegacion](#)



### Inicio:


Se muestran 4 planes próximos. Si estamos logueados, al clicar en uno de ellos nos llevará a la pantalla de detalles de ese plan. En caso de lo contrario nos llevará a la página de login. Debajo tenemos la barra de navegación que nos permite movernos a la página de crear plan, usuario o explorar planes.







7:05 [Battery Icon] [Signal Icon]


← Register








 Email


ENTER A VALID ERROR HERE

 Name

 Surname

 Wed Feb 10 2021

 Password

 Confirm Password

Refreshing...

## Registro:

En esta pantalla podemos registrar un usuario. Para ello tenemos que insertar los datos requeridos, aceptar los términos de uso y por último dar al botón de registrar.



 Email

 Password

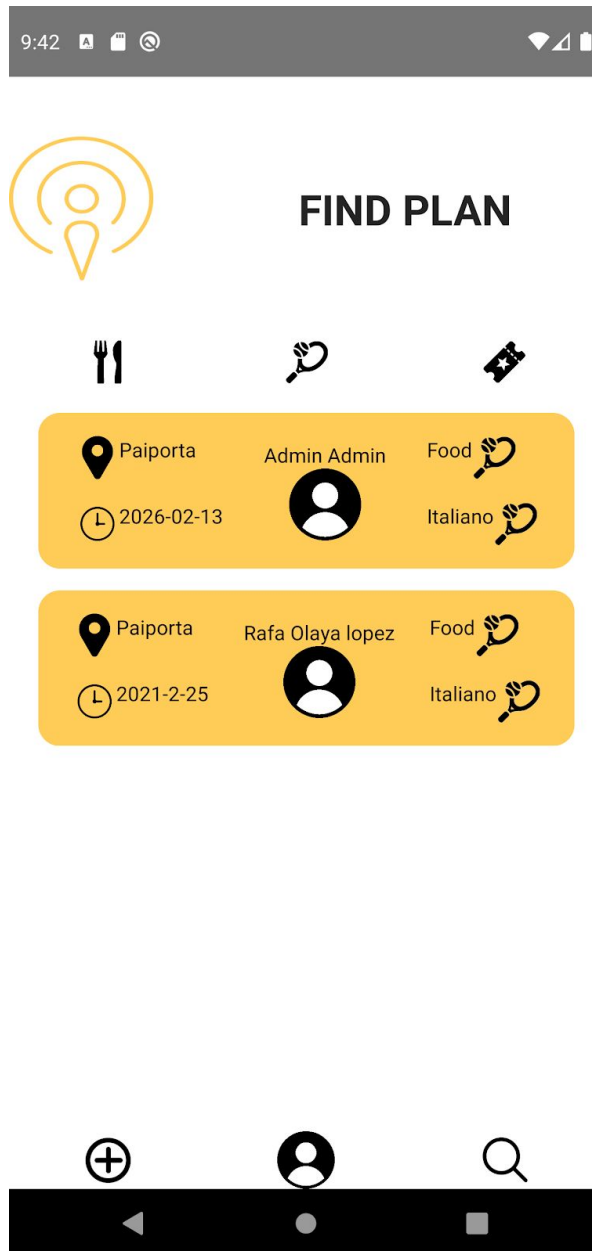
Login

Register

## Login:

Esta pantalla nos permite acceder si tenemos un usuario registrado introduciendo nuestro mail y nuestra contraseña. Luego daremos a Login. También podemos navegar a la página de registro si no tenemos usuario.





### Explore:

En esta pantalla podemos filtrar los eventos disponibles por categoría con los botones de arriba. Debajo tenemos la barra de navegación que nos deja transportarnos a la pantalla de crear planes o a la pantalla de usuario.





## PLAN DETAILS

test



Avenida los violes



2026-02-13



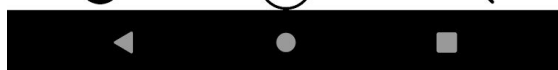
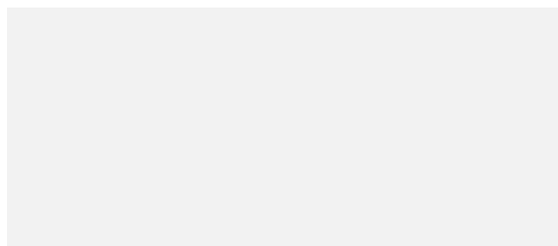
Italiano

### PLAN PEOPLE



Admin Admin

Confirm Assistance



### Detalles:

Esta pantalla nos muestra los detalles de un evento concreto. En ella, aparece una breve descripción, la localización del evento, la fecha, el subtipo del evento y la lista de usuarios apuntados a ese plan. También podemos unirnos a un plan, o salirnos si ya estamos apuntados.



**ToPlan**



## Documentación BackOffice:

**Ip:** <http://toplan.ml/> or <http://100.24.72.24:4000/>

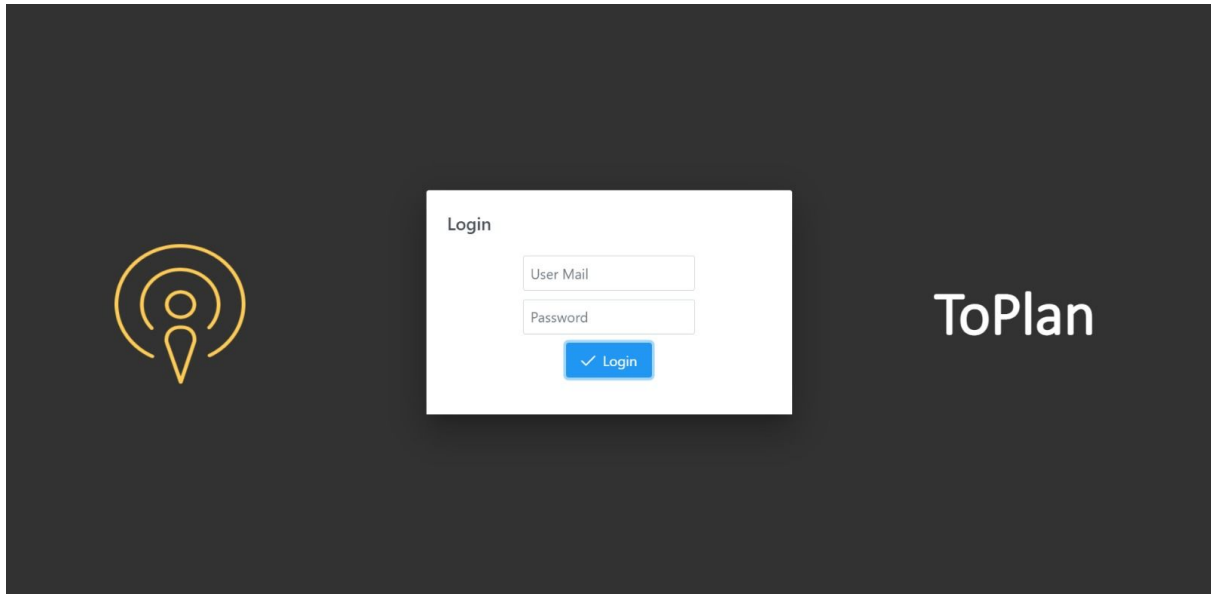
**Url Github:** [BackOffice\\_ToPlan-UserEvents](#)

### Screens:

- Login
- Users
- Events
- TypePlan

### Login

Esta Página se encarga de comprobar que el usuario que quiera acceder a la parte de la administración tanto de planes como users como tipo de planes. El Botón comprueba que el UserId existe y a su vez es admin, acto seguido comprobará que la password corresponde a el id de usuario introducido



- InputText Users {Adds to State id}
- InputText Password {Adds to State password}





-Button Accept Login {Check ifAdmin/CheckExistence of User and password and make visible UI}

## Users

Esta página se encarga de mostrar todas las acciones relacionadas con el manejo de usuarios

### Get All Users

Este TabPanel muestra una tabla con los datos de usuarios, los datos se cargan en el `onComponentDidMount()` o dándole al botón de Show

UsersEventsTypePlan					
Get	Get by Id	Get by Name	Update	Delete	Create
Get All User <input checked="" type="checkbox"/>					
Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

-Button Show {Display List of Users in DataTable}





## Get User by Id

Este TabPanel muestra una tabla con los datos del usuario, en caso de que introduzca un id en el inputText y le den al botón se mostrará en el DataTable el user con ese id.

UsersEventsTypePlan

GetGet by IdGet by NameUpdateDeleteCreate

Por IdUser Id

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserId {Adds to State id }
- Button Show {Display User in DataTable that Has that Id}



## Get User by Name

Este TabPanel muestra una tabla con los datos del usuario, en caso de que introduzca un name en el inputText y le den al botón se mostrará en el DataTable el user con ese name.

UsersEventsTypePlan

GetGet by IdGet by NameUpdateDeleteCreate

Por nombreUser Name

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserName {Adds to State name}
- Button Show {Display User in DataTable that Has that name}



## Update User

Este TabPanel muestra la tabla con los usuario existentes, en una primera instancia solo se mostrará un input el cual será el Id del User y un botón, si el Id introducido es correcto ( Existe en la base de datos) se desplegarán una serie de inputs y botones para cambiar todo lo relacionado con el user.

UsersEventsTypePlan

GetGet by IdGet by NameUpdateDeleteCreate

Update User

User Mail

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserId {Add to state id}
- Button check UserId {Check If UserID Exisist}



UsersEventsTypePlan

GetGet by IdGet by NameUpdateDeleteCreate

Update User

User Name

User SurName

User Password

birthDate

✓

Preferences

Sport☐Gastronomy☐Leisure☐

Make Admin

Remove Admin

Update user with id  
admin@gmail.com

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserName {Add to state name}
- InputText UserSurname {Add to state surname}
- InputText UserPassword {Add to state password}
- InputText UserBirthDate {Add to state BirthDate}
- CheckBox PreferencesSport {Add to state sport}
- CheckBox PreferencesGastronomy {Add to state Gastronomy}
- CheckBox PreferencesLeisure {Add to state Leisure}
- Button MakeAdmin {Makes Admin User}
- Button RemoveAdmin {Remove Admin User}
- Button UpdateUser {Change User}



## **Delete User**

Este TabPanel se encarga de pasandole un ID eliminarlo de la base de datos y haciendo su respectivo update en el dataTable que se mostrará en la parte de abajo

UsersEventsTypePlan

GetGet by IdGet by NameUpdateDeleteCreate

Delete User

User Mail

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserId {Add to state id}
- Button DeleteUser {Check User Exist and Remove it}



## Create User

Este TabPanel se encarga de comprobar que el ID pasado cumpla con los requisitos mínimos y de insertar todos los campos necesarios para la creación de un usuario

[UsersEventsTypePlan](#)

Get   Get by Id   Get by Name   Update   Delete   **Create**

Create User

User Mail

User Name

User SurName

User Password

birthDate

✓

Email	nombre	Surname	password	birthDate	Admin
admin@gmail.com	admin	istrador	12345	2000-02-13	✓

- InputText UserId     {Add to state Id}
- InputText UserName {Add to state Name}
- InputText UserSurname     {Add to state surname}
- InputText UserPassword     {Add to state Password}
- InputText UserBirthDate     {Add to state BirthDate}
- Button DeleteUser     {Check valid id and create User}



---

## Events

Esta página del BackOffice es la encargada así como la de users se encarga de la gestión y administración de usuarios, esta es la análoga a eventos. En todo momento se actualiza la DataTable inferior en cuanto le demos a los botones los cuales hacen una acción sobre la base de datos

### Update Event

Este TabPanel muestra la tabla con los usuario existentes, en una primera instancia solo se mostrará un input el cual será el Id del User y un botón, si el Id introducido es correcto ( Existe en la base de datos) se desplegarán una serie de inputs y botones para cambiar todo lo relacionado con el user.

- InputText EventId {Add to state EventId}
- Button CheckEvent {Check Event exist}
- InputMask EventDate {Add to state EventDate}
- InputText EventCity {Add to state City}
- InputText EventDirection {Add to state Direction}
- InputText EventProvince {Add to state Province}
- InputTextArea EventDescription {Add to state Description}
- InputText EventMembers {Add to state Members}
- Button UpdateEvent {Change the event}

### Update EventType

Este TabPanel muestra la tabla con los usuario existentes, en una primera instancia solo se mostrará un input el cual será el Id evento y un botón, si el Id introducido es correcto ( Existe en la base de datos) se desplegarán una serie de inputs y botones para cambiar todo lo relacionado con el evento

- InputText EventId {Add to state EventId}
- Button CheckEvent {Check Event exist}
- Dropdown TypePlanArray {Return TypePlanId and Subtype and Add to state TypePlanId}
- Button Update EventType {Change the Event Type}



UsersEventsTypePlan

Modificar Insertar Eliminar Get Events ListMembers

Update Event

EventId  ☒

Update EventType

EventId  ☒

EventId	Date	City	Direction	Province	Description	UserId	TypeId	Max Members
1	2021-2-20	Alaquas	calle 2 de mayo, 4	Valencia	Partido de futbol en el polideportivo de alaquas en la pista numero 3 hora: 17:00	rol19991@gmail.com	1	22

## Create Event

Este TabPanel se encarga de insertar todos los campos necesarios para la creación de un Evento

- InputText UserId {Add to state UserId}
- InputMask EventDate {Add to state EventDate}
- InputText EventCity {Add to state City}
- InputText EventDirection {Add to state Direction}
- InputText EventProvince {Add to state Province}
- InputTextArea EventDescription {Add to state Description}
- InputText EventMembers {Add to state Members}
- Dropdown TypePlanArray {Return TypePlanId and Subtype and Add to state TypePlanId}
- Button Insert Event {Post the Event}

UsersEventsTypePlan

Modificar **Insertar** Eliminar Get Events ListMembers

Insert Event

UserId  Date Event  Event Direction  EventCity  EventProvince

EventDescription

Event Members  Select a Subtype Plan

EventId	Date	City	Direction	Province	Description	UserId	TypeId	Max Members
1	2021-2-20	Alaquas	calle 2 de mayo, 4	Valencia	Partido de futbol en el polideportivo de alaquas en la pista numero 3 hora: 17:00	rol19991@gmail.com	1	22





## Delete Event

Este TabPanel se encarga de pasandole un ID eliminarlo de la base de datos y haciendo su respectivo update en el dataTable que se mostrará en la parte de abajo.

- InputText EventId {Add to state EventId}
- Button Delete Event {Delete the Event}

UsersEventsTypePlan

Modificar Insertar **Eliminar** Get Events ListMembers

Delete Event

EventId

Delete Event

EventId	Date	City	Direction	Province	Description	UserId	TypeId	Max Members
1	2021-2-20	Alaquas	calle 2 de mayo, 4	Valencia	Partido de futbol en el polideportivo de alaquas en la pista numero 3 hora: 17:00	rol19991@gmail.com	1	22

## Get Events

Este TabPanel muestra todos los eventos creados y dispone de un filtro en el cual, al seleccionar una categoría se muestran todos los datos pertenecientes a esta.

- Button Show All Events
- Dropdown TypePlanArray {Return TypePlanName and Subtype}
- Button FilterType {Shows in the table the events according to the selected type}

UsersEventsTypePlan

Modificar Insertar Eliminar **Get Events** ListMembers

Get All Events ☒ Filter by type: Select a Type Plan

EventId	Date	City	Direction	Province	Description	UserId	TypeId	Max Members
1	2021-2-20	Alaquas	calle 2 de mayo, 4	Valencia	Partido de futbol en el polideportivo de alaquas en la pista numero 3 hora: 17:00	rol19991@gmail.com	1	22



## Get List Members

Este TabPanel muestra la lista de miembros apuntados de cada evento creado.

- Button Show All Events
- Show the EventId and ListMembers of all Events in a DataTable

EventId	ListMembers
1	rol19991@gmail.com

## TypePlan

### Update TypePlan

Este TabPanel muestra la tabla con todos las categorías y subtipos, en una primera instancia solo se mostrará un input el cual será el Id Type y un botón, si el Id introducido es correcto ( Existe en la base de datos) se desplegarán una serie de inputs y botones para cambiar todo lo relacionado con el Type.

Update TypePlan

1 ☒

Type Name Type Subtype

Update Type

- InputText TypePlanId {Add to state TypePlanId}
- Button CheckEvent {Check Event exist}
- InputText TypeName {Add to state TypeName}
- InputText Subtype {Add to state Subtype}
- Button Update TypeEvent {Update the Event}



## **Create TypePlan**

Este TabPanel se encarga de insertar todos los campos necesarios para la creación de un Evento

The screenshot shows a web interface titled 'UsersEventsTypePlan'. At the top, there are four tabs: 'Modificar', 'Insertar' (which is highlighted), 'Eliminar', and 'Filtrar'. Below the tabs, the main content area is titled 'Insert TypePlan'. It contains two input fields: 'Type Name' and 'Type Subtype'. Below these fields is a blue button labeled 'Insert TypePlan'.

- InputText TypeName {Add to state TypeName}
- InputText Subtype {Add to state Subtype}
- Button Insert TypeEvent {Insert the TypePlan}

## **Delete TypePlan**

Este TabPanel se encarga de eliminar el TypePlan a partir de una id y posteriormente se actualiza y se muestra en un DataTable

The screenshot shows the same web interface as before, but the 'Eliminar' tab is now highlighted. The main content area is titled 'Delete Event'. It contains a single input field labeled 'Typeid'. Below this field is a blue button labeled 'Delete Type'.

- InputText TypePlanId {Add to state TypePlanId}
- Button Delete TypeEvent {Delete the Event}



## Get TypePlan

Este TabPanel muestra el Typeld, la categoría de cada tipo de evento y el subtipo de todos los TypePlan creados.

UsersEventsTypePlan		
Modificar	Insertar	Eliminar
Filtrar		
Typeld	Name	Subtype
1	sport	Futbol

{Show the TypePlan Id, TypeName, Subtype of all TypePlans in a DataTable}

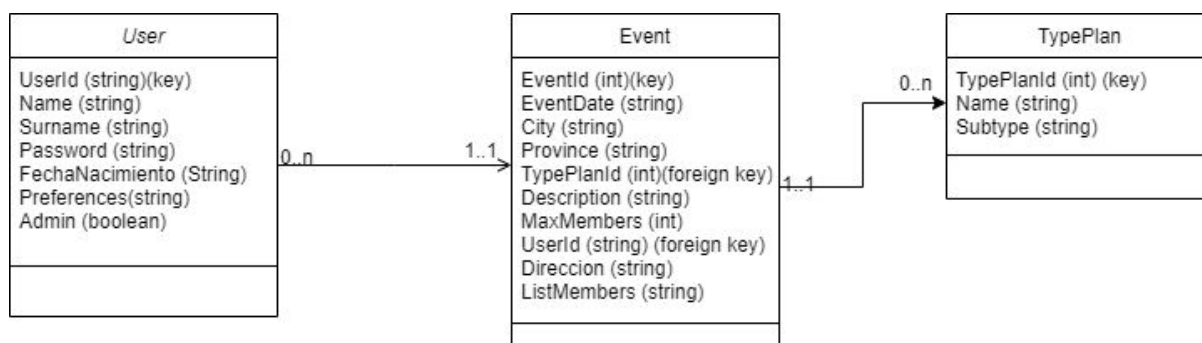


## Documentación API/BD

En la creación de la Web Api y la base de datos hemos utilizado la tecnología de EntityFramework. La IP en la que está alojada la API y la base de datos es la siguiente:  
**http://100.24.72.24:44360/**  
El código está en la rama “API” del github del proyecto.

### **BASE DE DATOS:**

Nuestra base de datos tiene 3 tablas relacionadas por varias foreign keys como se puede observar en el siguiente diagrama:



Explicación: Hemos creado la tabla “User” para poder almacenar toda la información necesaria sobre los usuarios. Hemos creado la tabla TypePlan para almacenar los tipos de planes que tenemos, guardando así la categoría y el subtipo. Por último tenemos la tabla Event creada para guardar toda la información sobre los eventos.  
Las relaciones entre las tablas son las siguientes:

#### **User 0..n 1..1 Event**

Hemos decidido este tipo de relación ya que opinamos que un usuario puede no estar asignado a un evento o a todos los que quiera. En cambio creemos que un evento solo puede tener, como máximo y mínimo, un usuario como creado.

#### **Event 1..1 0..n TypePlan**



**ToPlan**



Hemos decidido este tipo de relación ya que opinamos que un evento puede tener, como máximo y mínimo, un tipo. En cambio creemos que puede no existir ningún evento de un tipo o haber todos los que se quiera.

## WEB API

En la parte de la API hemos creado un controller, repository y clase principal para cada una de las tablas anteriores.

Los métodos que tenemos actualmente para poder acceder a nuestra base de datos son los siguientes:

### Users

#### ·GET

1.-

Ruta: `api/User/Check?id=(string)`

Retorno: Devuelve "true" si existe un usuario con la id proporcionada o "false" en caso contrario

Explicación: Método que usamos para comprobar que un usuario existe en nuestra base de datos o para no introducir un nuevo usuario con una id existente.

2.-

Ruta: `api/User1?id=(string)`

Retorno: Devuelve un "User" con el siguiente formato:

```
{
  "Name": "string ",
  "Email": "string",
  "FechaNacimiento": "string",
  "Preferences": "string"
}
```

Explicación: Método que utilizamos en la aplicación para recuperar los datos concretos de un usuario



**ToPlan**



3.-

Ruta: `api/User/GetAge?id=(string)`

Retorno: Devuelve la edad de un usuario concreto.

Explicación: Método utilizado para conseguir la edad de un usuario concreto

4.-

Ruta: `api/User/GetUserId?id=(string)`

Retorno: Devuelve un usuario con el siguiente formato:

```
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
}
```

Explicación: Método usado, principalmente, por el BackOffice. Devuelve un usuario con todos sus datos

5.-

Ruta: `api/User/GetUserName?n=(string)`

Retorno: Devuelve una lista con los usuarios que coincidan con el nombre aportado, con el siguiente formato:

```
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
},
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
```



**ToPlan**



```
    "Preferences": "string",  
    "Admin": bool  
}
```

Explicación: Método de filtrado de usuarios utilizado por el BackOffice

6.-

Ruta: api/User/CheckMail?m=(string)

Retorno: Devuelve "true" si el formato de mail es válido, "false" en caso contrario.

Explicación: Este método nos sirve para comprobar que el mail incorporado es válido.

Comentario: En un futuro se querría implementar una tabla en la base de datos con todos los mails que utilizamos, pudiendo modificar así los tipos de mails que se aceptarían.

7.-

Ruta: api/User/CheckAdmin?id=(string)

Retorno: Devuelve "true" si un usuario es Admin, "false" en caso de no serlo

Explicación: Método utilizado para dejar, o no, acceder a un usuario a la aplicación del BackOffice.

8.-

Ruta: api/User/GetPassword?id=(string)&p=(string)

Retorno: Devuelve "true" si la contraseña del usuario es válida, "false" en caso contrario.

Explicación: Método de comprobación de la contraseña de un usuario.

9.-

Ruta: api/UserDTO?id=(string)

Retorno: Devuelve un usuario con el siguiente formato:

```
{  
    "Name": "string",  
    "Surname": "string",  
    "FechaNacimiento": "string"  
}
```

Explicación: Método usado por la aplicación para recoger los datos necesarios de un usuario.



**ToPlan**





10.-

Ruta: api/Users

Retorno: Devuelve una lista con todos los usuarios en la base de datos con el siguiente formato:

```
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
},
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
}
```

Explicación: Método usado por la aplicación de BackOffice principalmente.

11.-

Ruta: api/User/Preferences?id="string"

Retorno: Devuelve una lista de "string" con las preferencias de un usuario.

Explicación: Método usado, tanto por la aplicación como por el BackOffice, para poder mostrar las preferencias de un usuario.

## POST

1.-

Ruta: api/User [FromBody]User

Explicación: Método para añadir un usuario con todos sus datos. El formato de entrada es el siguiente:

```
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
```



**ToPlan**



```
"Password": "string",  
"FechaNacimiento": "string",  
"Preferences": "string",  
}
```

2.-

Ruta: `api/User/WithoutPreferences [FromBody]User`

Explicación: Método para añadir un usuario sin insertar las preferencias. El formato de entrada es el siguiente:

```
{  
  "UserId": "string",  
  "Name": "string",  
  "Surname": "string",  
  "Password": "string",  
  "FechaNacimiento": "string",  
}
```

## PUT

1.-

Ruta: `api/User/AdminT?id=(string)`

Explicación: Añade permisos de administrador a un usuario

2.-

Ruta: `api/User/AdminF?id(string)`

Explicación: Quita permisos de administrador a un usuario

3.-

Ruta: `api/User/NewMail?id=(string)`

Explicación: Añade otro valor a la lista de mails permitidos.

Comentario: En un futuro se querría implementar una tabla en la base de datos con todos los mails que utilizamos, pudiendo modificar así los tipos de mails que se aceptarían.

4.-

Ruta: `api/User/Password?id=(string)&p=(string)`

Explicación: Modifica la contraseña de un usuario



**ToPlan**



5.-

Ruta: `api/User?id=(string)&n=(string)&s=(string)&f=(string)&p=(string)`

Explicación: Modifica los valores de un usuario

6.-

Ruta: `api/User/Preferences?id=(string)&p=(string)`

Explicación: Añade preferencias a un usuario

## **DELETE**

1.-

Ruta: `api/User?id=(string)`

Explicación: Borra a un usuario de la base de datos

## **Events**

### **GET**

1.-

Ruta: `api/Event/Check?id=(string)`

Retorno: Devuelve "true" si existe un evento con la id proporcionada, "false" en caso contrario

Explicación: Método para comprobar la existencia de un evento

2.-

Ruta: `api/Event/CheckType?id=(string)`

Retorno: Devuelve "true" si el tipo de evento es válido, "false" en caso contrario.

Explicación: Método para asegurarnos que la categoría del evento es válida.

Comentario: En un futuro se querría implementar una tabla en la base de datos con todos las categorías que utilizamos, pudiendo modificar así los tipos de categorías que se aceptarían.

3.-

Ruta: `api/Event/CheckUser?id=(string)&n=(string)`

Retorno: Devuelve "true" si un usuario está apuntado a un evento, "false" en caso contrario.

Explicación: Método utilizado por la aplicación para saber si el usuario actual está apuntado o no a un evento en concreto.



**ToPlan**



4.-

Ruta: api/Event

Retorno: Devuelve una lista con todos los eventos en la base de datos, con el siguiente formato:

```
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
},
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
}
```

Explicación: Método usado por el BackOffice para mostrar los datos de todos los eventos.



**ToPlan**



5.-

Ruta: `api/Event/List?id=(string)`

Retorno: Devuelve los datos de los usuarios que participan en el evento, con el siguiente formato:

```
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
},
{
  "UserId": "string",
  "Name": "string",
  "Surname": "string",
  "Password": "string",
  "FechaNacimiento": "string",
  "Preferences": "string",
  "Admin": bool
}
```

Explicación: Método usado para saber los datos de los usuarios de un evento en concreto

6.-

Ruta: `api/Event3`

Retorno: Devuelve una lista con 4 eventos próximos con el siguiente formato:

```
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "string",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string",
    "string"
  ]
}
```



**ToPlan**



```
]
},
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "string",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string",
    "string"
  ]
}
```

Explicación: Método usado por la página de inicio de la aplicación

7.-

Ruta: `api/Event/Type?p=(string)`

Retorno: Devuelve una lista con los eventos que coinciden con el subtipo puesto, con el siguiente formato:

```
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
},
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
```



**ToPlan**



```
"Description": "Comida amigos",
"MaxMembers": int,
"ListMembers": "string",
"Direccion": "string",
"UserId": "string",
"TypePlanId": int
}
```

Explicación: Método usado por el BackOffice para filtrar los eventos

8.-

Ruta: `api/Event/EventsUser?id=(string)`

Retorno: Devuelve una lista con los eventos en los que el usuario está apuntado, son el siguiente formato:

```
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
},
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
}
```

Explicación: Método usado por la aplicación para poder mostrar todos los eventos a los que un usuario está apuntado.



**ToPlan**



9.-

Ruta api/Event/Food

Retorno: Devuelve una lista con los eventos de tipo "food", con el siguiente formato:

```
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Food",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
},
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Food",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
}
```

Explicación: Método usado por la aplicación para recibir los eventos concretos con los datos necesarios.



**ToPlan**





10.-

Ruta api/Event/Sport

Retorno: Devuelve una lista con los eventos de tipo "sport", con el siguiente formato:

```
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Sport",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
},
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Sport",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
}
```

Explicación: Método usado por la aplicación para recibir los eventos concretos con los datos necesarios.



**ToPlan**



11.-

Ruta api/Event/Leisure

Retorno: Devuelve una lista con los eventos de tipo "leisure", con el siguiente formato:

```
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Leisure",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
},
{
  "EventId": int,
  "City": "string",
  "Date": "string",
  "Category": "Leisure",
  "Type": "string",
  "Name": "string",
  "Direccion": "string",
  "Description": "string",
  "MaxMembers": int,
  "listaMiembros": [
    "string"
  ]
}
```

Explicación: Método usado por la aplicación para recibir los eventos concretos con los datos necesarios.



**ToPlan**



12.-

Ruta: `api/Event1?id=(int)`

Retorno: Devuelve un evento en concreto con el siguiente formato:

```
{  
  "City": "string",  
  "Date": "string",  
  "Category": "string",  
  "Type": "string",  
  "Name": "string",  
  "EventId": int  
}
```

Explicación: Devuelve un evento con un formato concreto, lo usa principalmente la aplicación

13.-

Ruta: `api/Event2?id=(int)`

Retorno: Devuelve un evento con el siguiente formato:

```
{  
  "City": "string",  
  "Date": "string",  
  "Category": "string",  
  "Type": "string",  
  "Name": "string",  
  "Participants": int,  
  "MaxParticipants": int  
}
```

Explicación: Devuelve un evento con un formato concreto, necesario para la aplicación.



**ToPlan**



14.-

Ruta: api/Event4?id=(int)

Retorno: Devuelve un evento con el siguiente formato:

```
{
  "Description": "string",
  "Direction": "string",
  "EventDate": "string",
  "Type": "string",
  "Lista": [
    "string",
    "string"
  ]
}
```

Explicación: Devuelve un evento en un formato concreto para la aplicación

15.-

Ruta: api/Event/Type?id=(int)

Retorno devuelve la id del tipo del evento

Explicación: Método para recuperar el tipo de un evento.

16.-

Ruta: api/Eventtype?id=(string)

Retorno: Devuelve una lista según la categoría introducida. El formato es el siguiente:

```
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
  "Description": "Comida amigos",
  "MaxMembers": int,
  "ListMembers": "string",
  "Direccion": "string",
  "UserId": "string",
  "TypePlanId": int
},
{
  "EventId": int,
  "EventDate": "string",
  "City": "string",
  "Province": "string",
```





```
    "Description": "Comida amigos",
    "MaxMembers": int,
    "ListMembers": "string",
    "Direccion": "string",
    "UserId": "string",
    "TypePlanId": int
}
```

Explicación: Devuelve una lista con los eventos filtrados por tipo. Necesario para el filtrado del BackOffice

## POST

1.-

Ruta api/Event [FromBody] Event

Explicación: Inserta un evento en la base de datos. El formato es el siguiente:

```
{
    "EventDate": "string",
    "City": "string",
    "Province": "string",
    "Description": "string",
    "MaxMembers": int,
    "Direccion": "string",
    "UserId": "string",
    "TypePlanId": int
}
```

## PUT

1.-

Ruta: api/Event/AddUser?id=(int)&n=(string)

Explicación: Añade a un usuario a un evento

2.-

Ruta: api/Event/RemoveUser?id=(int)&n=(string)

Explicación: Borra a un usuario de la lista de un evento

3.-

Ruta: api/Event/Type?id=(int)&t=(int)

Explicación: Actualiza el tipo del evento



**ToPlan**



4.-

Ruta: `api/Event/AddType?id=(string)`

Explicación: Añade un tipo de evento válido

Comentario: En un futuro se querría implementar una tabla en la base de datos con todos las categorías que utilizamos, pudiendo modificar así los tipos de categorías que se aceptarían.

## DELETE

1.-

Ruta: `api/Event?id=(int)`

Explicación: Borra un evento según su id

## Typeplan

### GET

1.-

Ruta: `api/TypePlan/List`

Devuelve una lista con todos los tipos de planes, el formato es el siguiente:

```
{
  "TypePlanId": int,
  "Name": "string",
  "Subtype": "string"
},
{
  "TypePlanId": int,
  "Name": "string",
  "Subtype": "string"
}
```

Explicación: Método usado por el BackOffice principalmente

2.-

Ruta: `api/TypePlan/Check?id=(int)`

Retorno: Devuelve "true" si existe un tipo con esa id, "false" en caso contrario

Explicación: Método usado por el BackOffice para comprobar la existencia de un Typeplan



**ToPlan**



3.-

Ruta: `api/TypePlan/ListDTO`

Retorno: Devuelve una lista con todos los tipos de planes con un formato concreto:

```
{
  "label": "string",
  "value": int
},
{
  "label": "string",
  "value": int
}
```

Explicación: Devuelve una lista espacial que utiliza la aplicacion

## POST

1.-

Ruta: `api/TypePlan [FromBody] TypePlan`

Explicación: Inserta un TypePlan nuevo. El formato es el siguiente:

```
{
  "Name": "string",
  "Subtype": "string"
}
```

## PUT

1.-

Ruta: `api/TypePlan?id=(int)&n=(string)&s=(string)`

Explicación: Modifica los datos de un plan

## DELETE

1.-

Ruta: `api/TypePlan?id=(int)`

Explicación: Borra un TypePlan según su id



**ToPlan**



---

## **Propuestas de mejora:**

En general estamos bastante contentos con nuestro trabajo y lo que nos ha dado tiempo a hacer. Aún así tenemos varias propuestas de mejoras y cosas que en general no han dado tiempo a hacer.

- Hay varias cosas comentadas en la documentación de la API y la base de datos. Entre ellos tenemos la creación de dos tablas extras para guardar los tipos de mails que se aceptan y los tipos de categorías que se aceptan.

- En general más cosas relacionadas con las preferencias. Era una parte que nos gustaba mucho, queríamos implementar métodos que nos devolvieran eventos basados en las preferencias de cada uno de los usuarios, pero por tema de tiempo no ha podido ser.

- Temas de seguridad y encriptación. Tenemos preparado, tanto en la aplicación como en el BackOffice, variables donde guardaremos el token si nos hubiese dado tiempo a modificar la API y poder añadir estos temas. También hubiésemos querido implementar el tema de encriptar las contraseñas.

- Temas de diseño. Nos hubiese gustado mejorar el tema del diseño de la interfaz.

- Queríamos añadir diferentes avatares para que cada persona puede elegir uno o permitir subir una foto y usarla como avatar.

Todas las cosas anteriores serán implementadas en próximas actualizaciones, en caso de seguir adelante con el proyecto.







---

## **Conclusión:**

Ha sido un proyecto muy interesante, por desgracia, debido a la carencia de tiempo, no hemos podido incorporar y pulir todo lo que hubiésemos querido el proyecto. Aún así creo que hemos logrado un resultado correcto. A continuación cada uno de los integrantes escribirá un comentario personal sobre el proyecto.

### **Adrián Javier Ballesteros Romero:**

Ha sido un proyecto muy interesante. Personalmente me he dedicado casi todo el tiempo a la parte que más me gusta, que es el BackEnd, pero también he estado ayudando a los que trabajaban en otros campos y me ha servido para aprender y familiarizarme con temas en los que normalmente no estoy tan a gusto. También me ha gustado la experiencia de trabajar en equipo como si fuéramos una pequeña empresa y ver las dificultades que esto puede producir. Ha sido una pena todo el tema del Covid-19 que ha dificultado mucho el trabajo en persona, pero creo que hemos podido superarlo bastante bien. En general ha sido una buena experiencia.

### **Ignacio Pedrós Gil:**

El desarrollo del proyecto ha sido toda una experiencia. El hecho de realizar este proyecto en el periodo marcado, ha sido todo un reto personalmente, hemos tenido bastantes dificultades que hemos ido solventando por el camino. Me he dedicado a trabajar en el área de Frontend realizando el BackOffice, la parte de administración del proyecto. Me he sentido muy bien durante la realización del proyecto, hemos trabajado conjuntamente todos con todos apoyándonos en todo momento y me siento satisfecho por todo el trabajo que he realizado. He estado bastante cómodo con todos los componentes del grupo y ha sido muy enriquecedora la experiencia de trabajar en grupo. Por último, he aprendido muchísimo con este trabajo y me ha servido para crecer, tanto individualmente como profesionalmente.

### **Adrián Pérez González:**

Al principio un poco de caos, no teníamos claro por dónde empezar el primer día y nada y una vez superada esa mini crisis todo ha ido sobre ruedas, he realizado pantallas de React-Native y diseño en Axure, además de diseño gráfico y el tema de economía del proyecto, la organización ha sido buena puesto que era el Scrum Master y Adrian J me ha apoyado también organizando. Por mi parte encantadísimo con el equipo que me ha tocado y una buena experiencia. Una pena que el entorno COVID-19 limite el trabajar en grupo en clase porque habría sido genial.



**ToPlan**



---

**Rafa Olaya López:**

Proyecto muy interesante con el que hemos podido trabajar aspectos más relacionados con el trabajo en la empresa aplicando metodologías como Scrum que nos han hecho trabajar mejor, creo que ha sido un proyecto interesante que con algo más de tiempo podría haber sido una aplicación muy útil ya no hay muchas opciones en el mercado. Hemos podido desarrollar conceptos que no hemos visto en clase y aprenderlos por cuenta propia haciéndonos más autodidactas a la hora de aprender.

**Jorge Miguel Cervera Gómez:**

Sinceramente me ha parecido un proyecto el cual pienso que con más tiempo y recursos podría haber salido una muy buena aplicación y con aras de futuro. Personalmente me he encargado junto a mi compañero Nacho de la realización del BackOffice para la gestión de eventos users y tipos de plan, aun así me he sentido muy útil en el sentido de que nos hemos apoyado todos mutuamente en cualquier error o cualquier problema. Para concluir esta experiencia me ha sido muy útil tanto a la adquisición de nuevos conocimientos y como crecimiento personal ya que he experimentado lo que realmente me voy a encontrar cuando salga al mundo laboral como desarrollador.





**ToPlan**