

# **MAGIC PABS APP**

# Inidce

1) ABSTRACT.....	3
2) INTRODUCCIÓN.....	3
2.1) OBJETIVO.....	3
2.2) JUSTIFICACIÓN.....	3
2.3) ANÁLISIS DE LO EXISTENTE.....	4
3) NECESIDADES EMPRESARIALES.....	4
3.1) RECURSOS HUMANOS.....	4
3.2) CONTRATOS.....	4
3.3) OBLIGACIONES EN MATERIA DE SEGURIDAD SOCIAL.....	4
3.4) PREVENCIÓN DE RIESGOS LABORALES.....	4
3.5) MEDIDAS PREVENTIVAS.....	5
3.6) FORMA JURÍDICA.....	5
3.7) TRAMITES ADMINISTRATIVOS.....	5
4) REQUISITOS FUNCIONALES DE LA APLICACIÓN.....	5
5) DISEÑO Y ANÁLISIS.....	6
5.1) DIAGRAMA DE LA ARQUITECTURA.....	6
5.2) DISEÑO DE DATOS.....	6
5.3) DIAGRAMA DE CASOS DE USO.....	7
5.3.1) Diagrama de casos de uso en la aplicación de Windows.....	7
5.3.2) Diagrama de casos de uso en la aplicación de Android.....	8
5.4) DIAGRAMA DE CLASES.....	9
6) CODIFICACIÓN.....	9
6.1) LENGUAJES Y ENTORNOS DE DESARROLLO USADOS.....	9
7) MANUAL DE USUARIO.....	10
7.1) MANUAL DE LA APLICACIÓN DE WINDOWS.....	10
7.1.1) CLIENTES.....	10
7.1.2) ESPECTÁCULOS.....	12
7.1.3) EVENTOS.....	14
7.2) MANUAL DE LA APLICACIÓN DE ANDROID.....	17
7.2.1) CLIENTES.....	17
7.2.2) ESPECTÁCULOS.....	18
7.2.3) EVENTOS.....	19
8) REQUISITOS E INSTALACIÓN.....	20
8.1) DESCRIPCIÓN DEL ENTREGABLE.....	20
8.1.1) WINDOWS.....	20
8.1.2) ANDROID.....	20
8.1.3) API REST.....	21
8.2) REQUISITOS:.....	21
8.3) INSTALACIÓN DE LA API REST EN UN SERVIDOR LINUX.....	22
8.4) INSTALACIÓN DE LA APLICACIÓN DE ANDROID.....	23
8.5) INSTALACIÓN DE LA APLICACIÓN DE WINDOWS.....	23
9) TESTING Y ANÁLISIS DE RENDIMIENTO.....	24
9.1) TEST DE CARGA.....	24
9.2) TEST DE ESCALABILIDAD.....	25
9.3) TEST DE PICOS.....	27
9.4) TEST DE ESTRÉS.....	28
10) CONCLUSIONES.....	29
10.1) CONCLUSIONES SOBRE EL TRABAJO REALIZADO.....	29
10.2) POSIBLES AMPLIACIONES Y MEJORAS.....	30
11) BIBLIOGRAFÍA.....	30

# 1) ABSTRACT

Magic Pabs App is the solution every magician needs to manage his shows. Magic Pabs App offers an easy way to manage magic shows, usually most of magicians use a spreadsheet to manage their shows and their clients, but with this app you could manage all your shows and clients from your computer and if you aren't at home you can use the mobile app to watch all the info you have submitted with your computer.

The computer app is a simple and intuitive app where you can manage everything, you can add and edit new clients, you can manage your shows by having the most important information about your shows like the title, a brief description of the show, the type of magic that you're gonna use on that show and the duration of the show. It also has the part where you can manage the events you have, like for example a wedding where they hired you to do a magic show, you can have very important information about those events, like the client who has hired you, the show you're gonna do, the location of the event, and the date of the event.

The android app will make your shows easier with a portable version of the computer app where you can watch all the information of your clients, shows and events.

The apps will be interconnected with a MySQL database to keep all the data, all the data will be accessed and served to the apps by an API REST on a Linux server. The Windows app will be developed with WPF in C# and the android app will be developed in Kotlin.

To sum up with Magic Pabs App you will manage all your shows in an easy and intuitive way and it's an app that will be revolutionary for magicians because it's the first app of this type. I guess this app could help a lot of magicians and could encourage other developers to start developing apps for magicians.

## 2) INTRODUCCIÓN

### 2.1) OBJETIVO

El objetivo de este proyecto consiste en el desarrollo de una serie de aplicaciones que servirán para que un ilusionista pueda llevar a cabo la organización de sus espectáculos de la forma mas cómoda y sencilla posible.

### 2.2) JUSTIFICACIÓN

La razón por la que he decidido hacer este proyecto es por la ausencia de aplicaciones que puedan gestionar espectáculos de magia de manera sencilla, la mayoría de magos utilizan para esto hojas de cálculo lo que lo hace bastante ineficiente a la hora de tener una cantidad bastante grande de clientes. La finalidad de este proyecto como ya he comentado es la de solucionar el problema de escasez de este tipo de aplicaciones y ademas hacerlo de una forma cómoda y sencilla para el ilusionista.

## **2.3) ANÁLISIS DE LO EXISTENTE**

El proyecto es un proyecto muy innovador ya que no existe ninguna aplicación similar en el mercado, además que innova bastante en el tema de que el mundo del ilusionismo reniega un poco de las nuevas tecnologías, ya que sin contar un par de aplicaciones con las que se pueden hacer un par de trucos de magia, no hay apenas aplicaciones que puedan ayudar a los ilusionistas, a sí que una aplicación que pueda ayudar a los ilusionistas puede ser un empujón para empezar a hacer más software para ilusionistas.

## **3) NECESIDADES EMPRESARIALES**

### **3.1) RECURSOS HUMANOS**

La empresa encargada de desarrollar es una empresa muy pequeña, donde solo trabajan 6 personas, de las cuales 5 se dedican exclusivamente a programar, uno se dedica a la programación de la base de datos y la API REST, otro se dedica a la programación de los layouts de la aplicación de Android, otro se dedica a programar todo el funcionamiento interno de la aplicación de Android, luego tenemos a otro dedicándose a la programación de los layouts de la aplicación de Windows y el último se dedica a la programación del funcionamiento interno de la aplicación de Windows.

Por último tendríamos al jefe de proyecto que se dedicaría a el diseño de todo el proyecto, el contacto con el cliente y a toda la parte del testing.

### **3.2) CONTRATOS**

Al ser una empresa bastante nueva creada por desarrolladores recién salidos de acabar sus estudios los contratos de los programadores son contratos indefinidos como junior cobrando un sueldo de 1300 brutos mensuales trabajando 40 horas semanales y con 31 días de vacaciones anuales.

Luego el contrato del jefe de proyecto es un contrato indefinido con un sueldo de 1800 brutos mensuales trabajando 40 horas semanales y con 31 días de vacaciones anuales.

### **3.3) OBLIGACIONES EN MATERIA DE SEGURIDAD SOCIAL**

Los trabajadores anteriormente nombrados estarán dados de alta en el régimen de la seguridad social. También hemos registrado nuestra empresa en el registro mercantil como sociedad limitada.

### **3.4) PREVENCIÓN DE RIESGOS LABORALES**

Los posibles riesgos laborales que podremos encontrar en esta empresa son:

- Problemas de espalda por malas posturas a la hora de trabajar.
- Problemas de vista por una larga exposición a pantallas.
- Problemas en las muñecas por una mala posición al usar el teclado del ordenador.

### **3.5) MEDIDAS PREVENTIVAS**

Para solucionar los problemas de espalda las oficinas de la empresa disponen de sillas ergonómicas para poder mantener una buena postura en ellas mientras se trabaja.

Para prevenir los problemas de vista se proporcionan unas gafas especiales para prevenir estos problemas, además de ejercer descansos cada cierto tiempo para poder descansar la vista.

Para prevenir problemas en las muñecas la oficina cuenta con reposa muñecas adecuados además de teclados y ratones ergonómicos para no tener las muñecas en posiciones antinaturales que puedan generar este tipo de problemas.

### **3.6) FORMA JURÍDICA**

La empresa se ha registrado como sociedad limitada como hemos mencionado anteriormente, la razón por la que se ha decidido registrar la empresa de dicha manera es porque no es necesaria una gran suma de dinero para poder empezarla, tan solo con 3000€ ya bastaría para registrar la empresa y porque solamente se necesitaría un socio para poder registrar la empresa.

### **3.7) TRAMITES ADMINISTRATIVOS**

Lo primero fue comprobar que el nombre de nuestra empresa no estuviese registrado en el registro mercantil y registrarlo nosotros.

Después contratamos a una gestora que se encargó de preparar los estatutos de la empresa, solicitar el CIF provisional y pagar los impuestos de gestión ante el registro mercantil.

Después pagamos los impuestos de transmisiones patrimoniales y actos jurídicos documentados y por último nos registramos en el registro mercantil de nuestra provincia.

## **4) REQUISITOS FUNCIONALES DE LA APLICACIÓN**

La aplicación intenta solucionar la manera de gestionar espectáculos de magia ya que actualmente la única manera que hay de hacer esto es mediante hojas de cálculo, lo que lo hace completamente inviable sobretodo en cuanto la cantidad de espectáculos y clientes que tienes es medianamente grande. Por esa razón una aplicación desde la cual se pueda gestionar todo esto de manera fácil e intuitiva es muy útil, por eso esta aplicación cuenta con todo lo necesario, el ilusionista contará con una aplicación de escritorio para Windows desde donde podrá gestionarlo todo. Tendrá un apartado de clientes desde el cual podrá tanto ver los clientes que ya han solicitado los servicios del ilusionista como podrá añadir nuevos clientes que hayan solicitado sus servicios. Por otro lado tendremos otro apartado desde el cual se podrán gestionar todos sus espectáculos, este apartado contará con una parte desde la cual se podrá ver toda la información de los espectáculos como el título, la duración del mismo y una descripción con la información clave del espectáculo, además contará con opciones para agregar y modificar nuevos espectáculos. Por último, contará con una última sección desde la cual se gestionaran los eventos, tendrá, como todas las secciones anteriores, una parte desde donde se pueden visualizar los datos de los eventos, como por ejemplo la fecha de los eventos, la dirección de donde se va a celebrar dicho evento, el espectáculo que vas a ofrecer y el cliente que te lo ha solicitado.

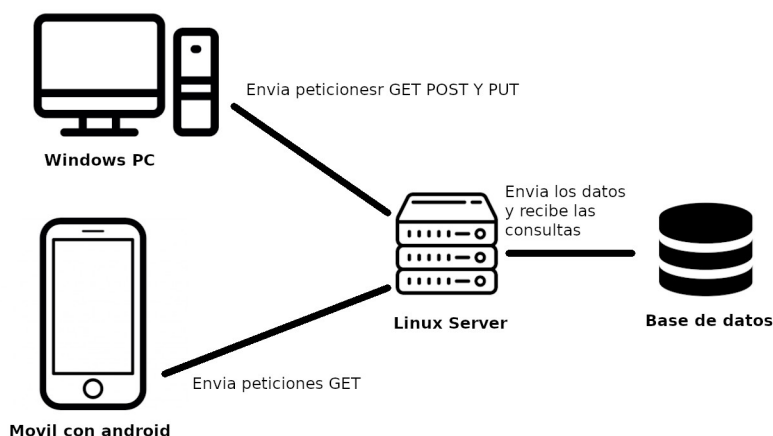
En cuanto a la aplicación de móvil Android contará con las mismas partes que la aplicación de Windows pero estas solo permitirán la visualización de los datos y algunas otras funciones propias de una aplicación de móvil como la posibilidad contactar con los clientes.

Estas aplicaciones serán usadas solamente por el ilusionista y su equipo puesto que contienen datos de personas reales el uso de estas aplicaciones queda limitado al ilusionista y en el caso de tener un equipo detrás que le ayude con la gestión de los espectáculos también tendrán acceso a estas aplicaciones.

Al estar esta aplicación dirigida a personas que puede que no estén familiarizadas con las nuevas tecnologías la aplicación se ha diseñado para ser lo mas intuitiva y accesible posible para que cualquier ilusionista sea capaz de usarla sin tener grandes conocimientos con nuevas tecnologías.

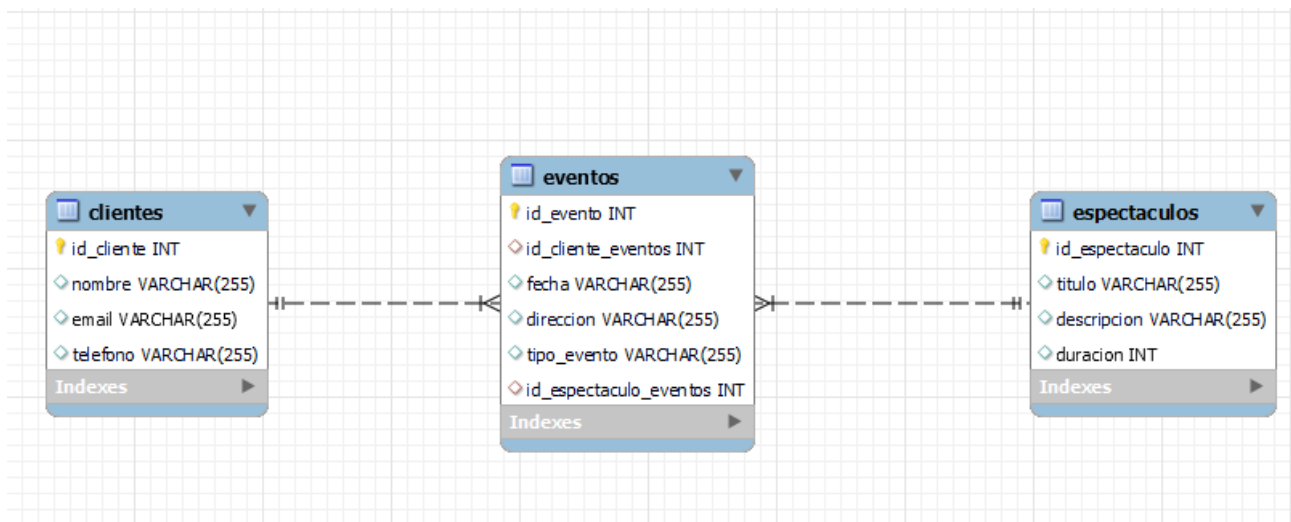
## 5) DISEÑO Y ANÁLISIS

### 5.1) DIAGRAMA DE LA ARQUITECTURA



Para poder implantar todo el producto necesitaríamos un servidor con Linux donde instalar nuestra base de datos y nuestra API REST, para eso el servidor deberá tener instalado el MySQL y el Wildfly, también necesitaríamos un PC con Windows donde instalaremos la aplicación de escritorio y un dispositivo móvil con Android.

### 5.2) DISEÑO DE DATOS



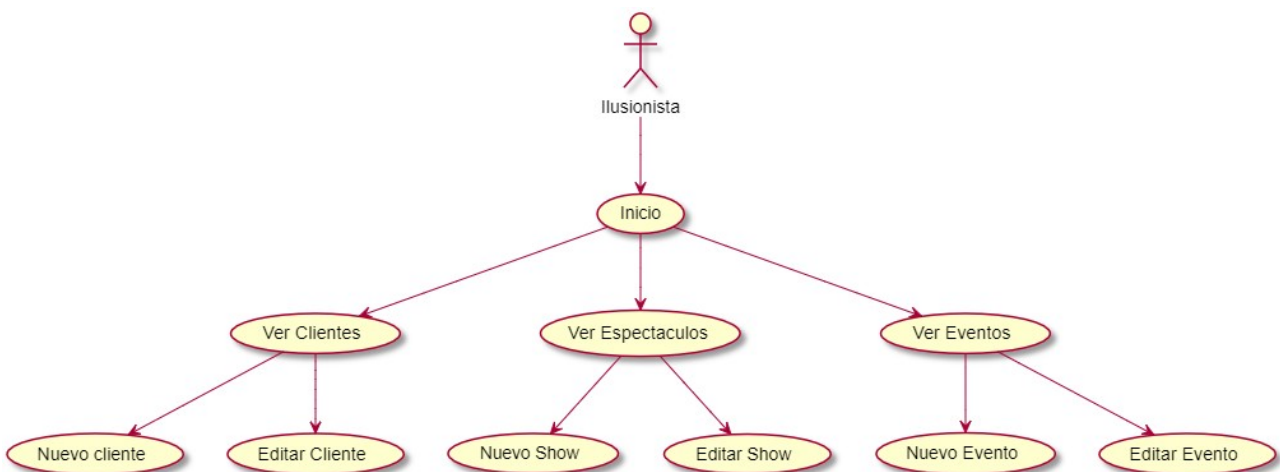
-La tabla de clientes contiene toda la información de los clientes que han solicitado los servicios del ilusionista, desde el nombre hasta los datos de contacto.

-La tabla de espectáculos contiene la información de los espectáculos que dispone el ilusionista para su contratación, contiene el título del espectáculo, una breve descripción donde se pondrá un pequeño resumen de como será el espectáculo.

-La tabla eventos contiene la información de los eventos que los clientes han contratado, por esa razón se relaciona con la tabla clientes para saber que cliente ha contratado que evento, además también se relaciona con la tabla de espectáculos para saber que espectáculo se realizara en dicho evento, además contendrá información básica del evento como es la fecha, la dirección y el tipo de evento del que se trata (bodas, bautizos, cumpleaños...).

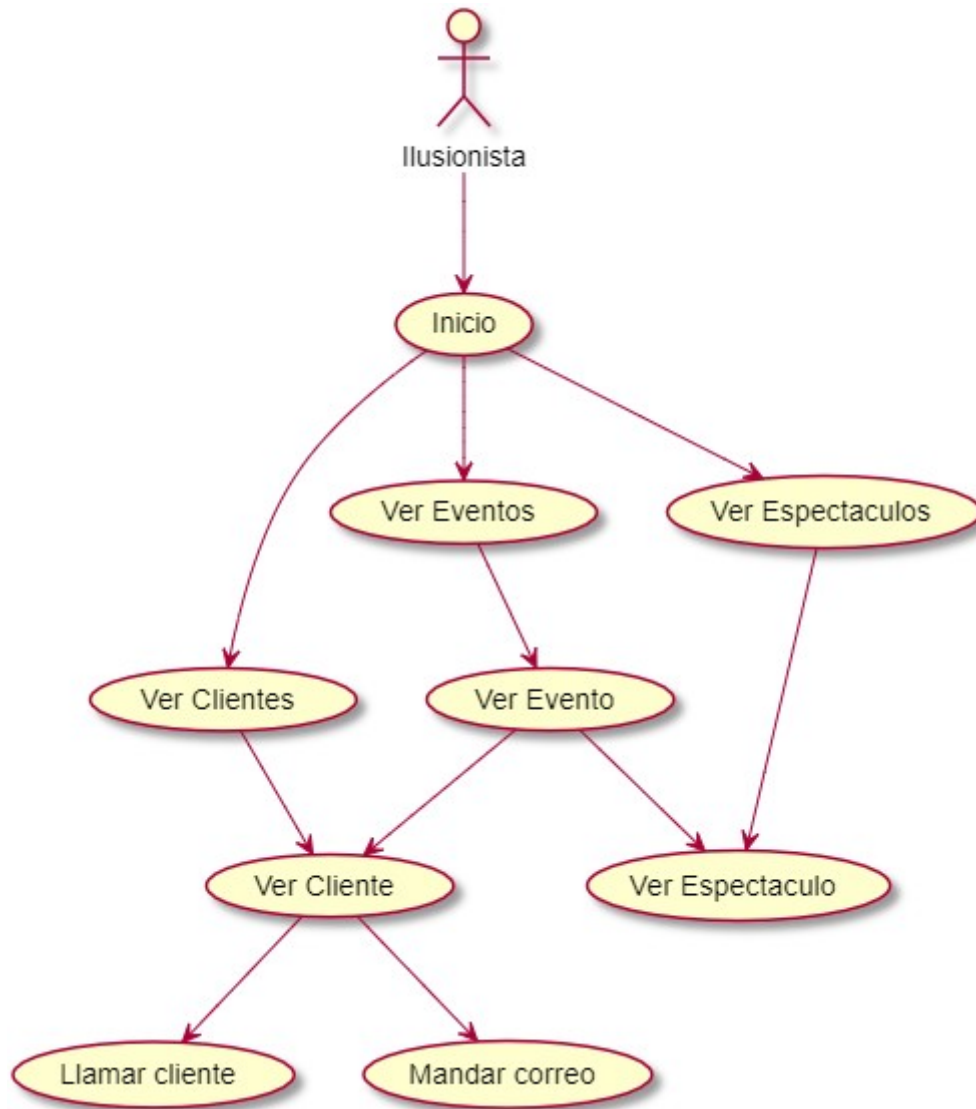
## 5.3) DIAGRAMA DE CASOS DE USO

### 5.3.1) Diagrama de casos de uso en la aplicación de Windows



En esta aplicación como se puede observar, el ilusionista, que es el usuario de la aplicación puede acceder nada mas entrar a la aplicación a una pantalla de inicio con el logotipo del ilusionista, desde esta se podrá acceder mediante un menú al resto de pantallas desde la que visualizar la información, desde las cuales se podrá acceder a sus respectivas pantallas de creación y edición

### 5.3.2) Diagrama de casos de uso en la aplicación de Android



En la aplicación de Android nada mas empezar, al igual que en la de Windows, se puede ir a una pantalla de inicio con el logotipo del ilusionista, desde esa pantalla y mediante un menú se podrá acceder a una serie de pantallas que listan tanto los clientes como los eventos y los espectáculos.

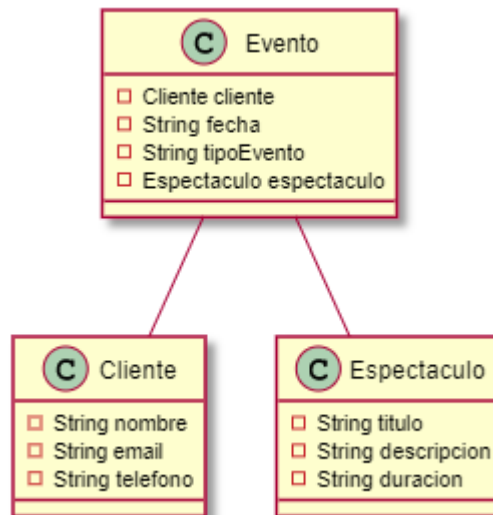
En cada pantalla se podrá seleccionar un cliente, evento o espectáculo específico dependiendo de en que pantalla de visualización estemos.

Desde la pantalla de ver cliente, tendremos dos opciones, la de llamar al cliente y la de mandarle un correo.

Por último desde la pantalla de ver evento podremos ir a la pantalla de ver cliente o a la de ver espectáculo, mostrándonos así el contenido de ese evento.



## 5.4) DIAGRAMA DE CLASES



El diagrama de clases de nuestras aplicaciones es bastante similar a la estructura de la base de datos ya que es de esta de quien recibe los datos con los que va a trabajar. Se basa en 3 clases de datos, tendríamos la clase de cliente, que sirve para agrupar la información importante sobre los clientes como son el nombre, su teléfono y su correo electrónico. Por otro lado tendríamos la clase de espectáculo, que contiene la información del espectáculo que vayamos a hacer, como es su título, su descripción y su duración. Por último tendríamos la clase de Evento, que es la clase que relaciona las dos clases anteriores, ya que contiene todos los datos del evento, el cliente que solicita el espectáculo, el espectáculo solicitado, el tipo de evento que se va a realizar y la fecha del espectáculo.

## 6) CODIFICACIÓN

### 6.1) LENGUAJES Y ENTORNOS DE DESARROLLO USADOS

Para la API REST decidimos usar JAVA en un servidor WildFly desplegado en un servidor Linux en azure. Para desarrollar esto usamos el entorno de desarrollo Apache NetBeans.

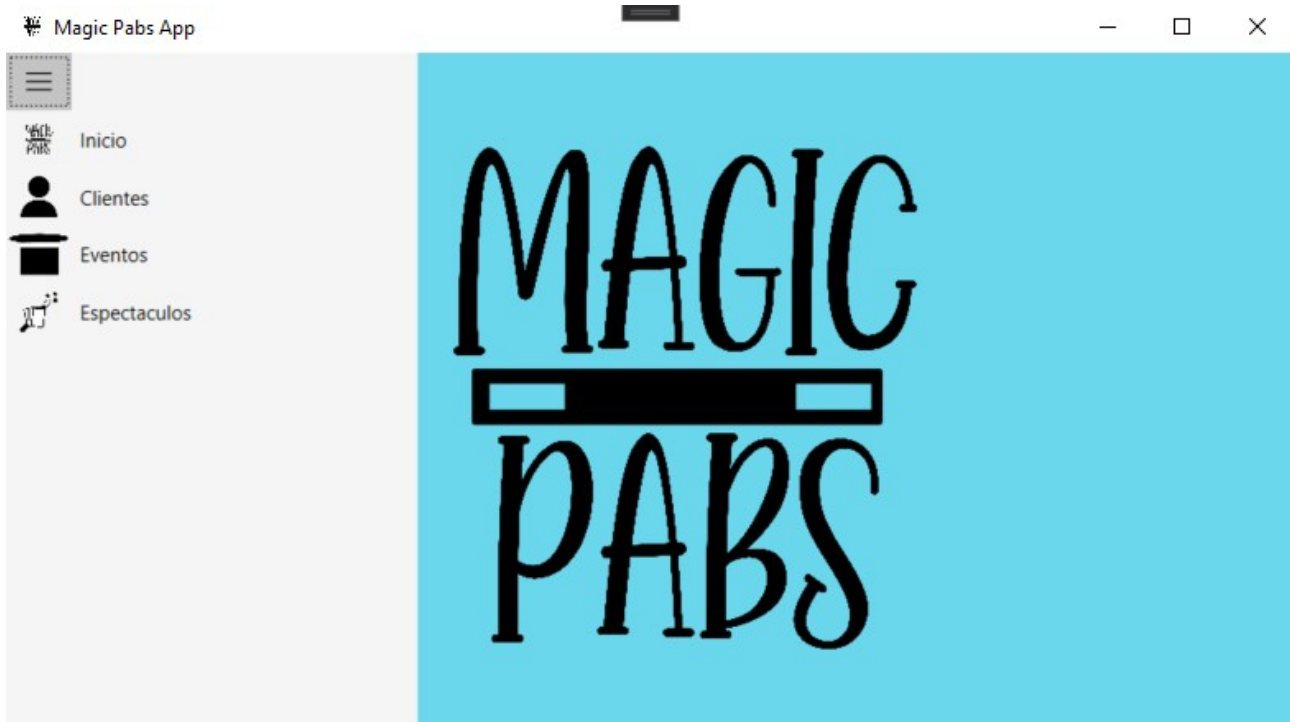
Para la aplicación de Android se ha usado Kotlin ya que es el lenguaje con el que mejor funcionan las aplicaciones nativas para Android. Para llevar a cabo el desarrollo de la app de android usamos Android Studio.

Y para la aplicación de Windows se ha usado C# con el framework para el desarrollo de interfaces WPF. Para el desarrollo de la aplicación de Windows usamos Visual Studio 2019.

## 7) MANUAL DE USUARIO

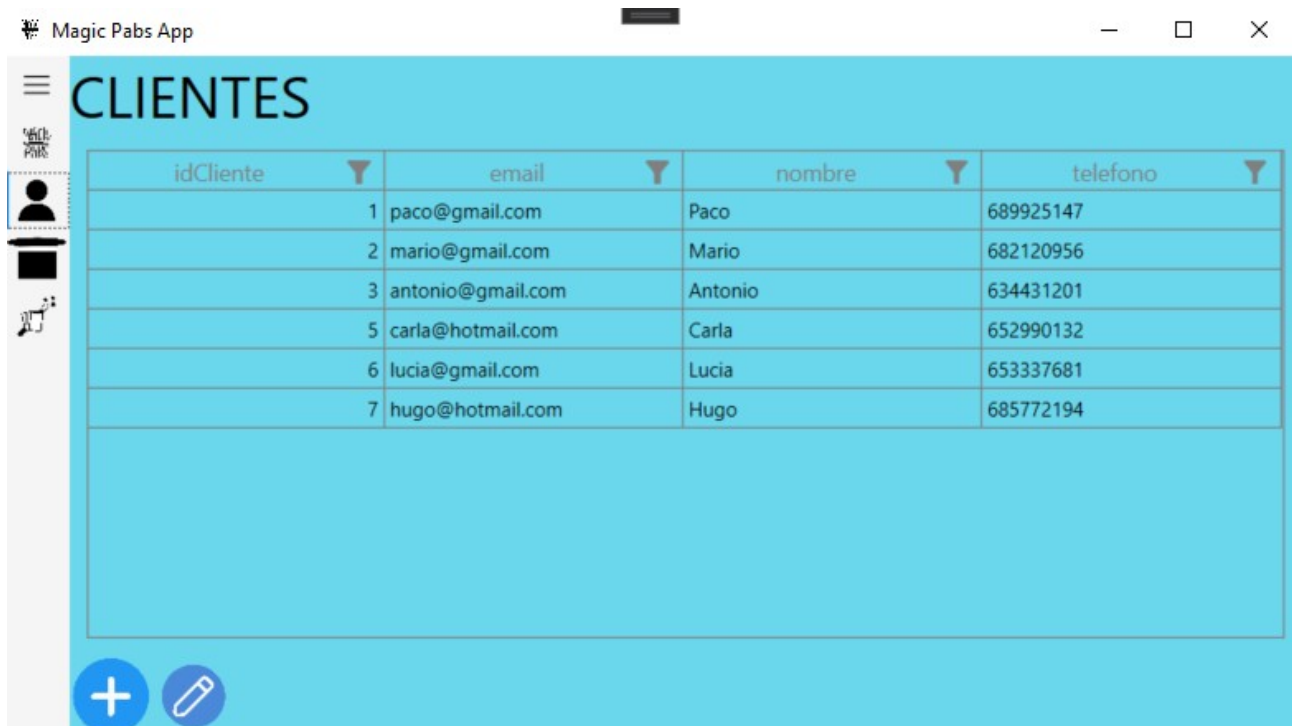
### 7.1) MANUAL DE LA APLICACIÓN DE WINDOWS

Nada mas entrar a la aplicación nos encontraremos con una página de inicio con el logotipo del ilusionista desde la cual podremos acceder al menú para poder navegar cómodamente por la aplicación.

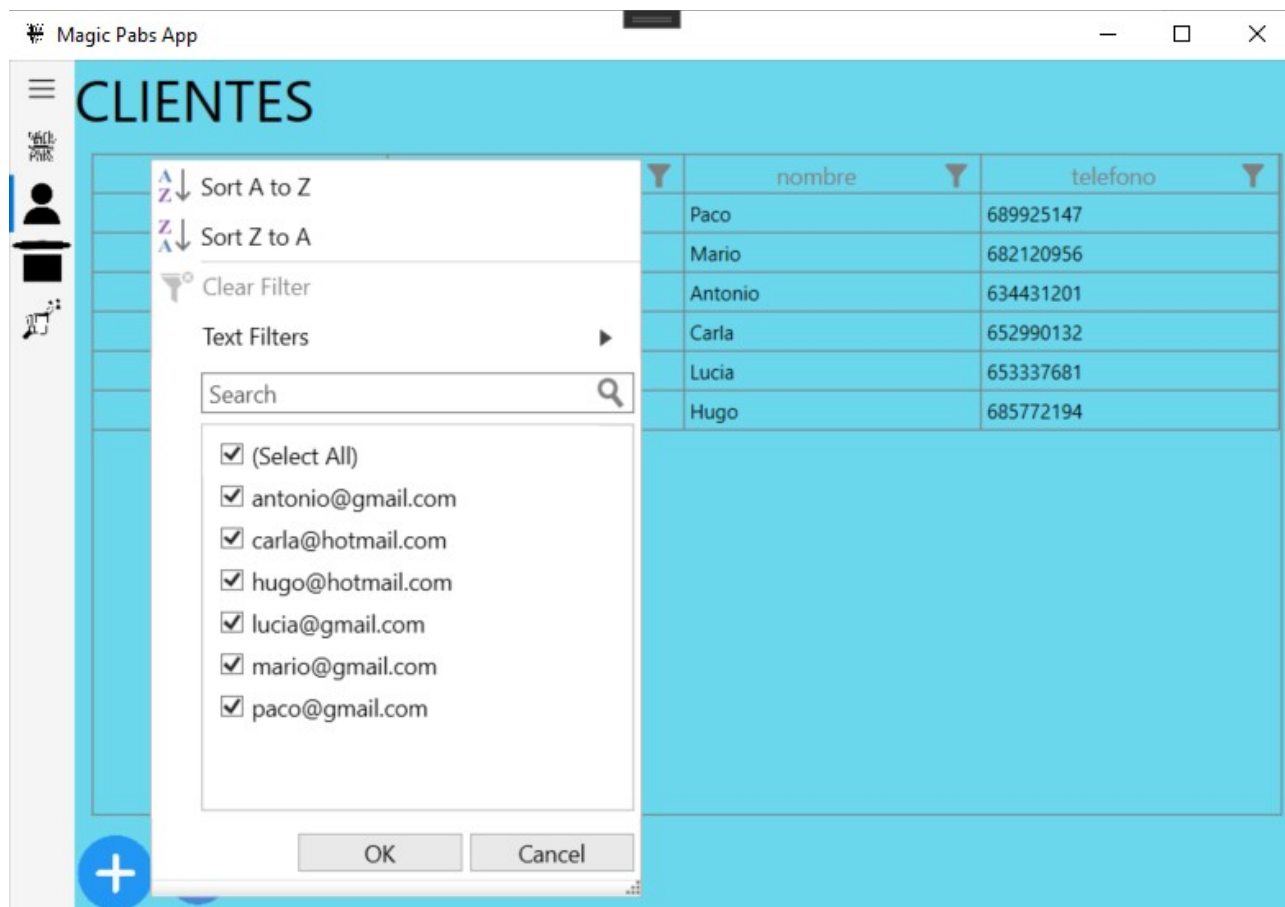


#### 7.1.1) CLIENTES

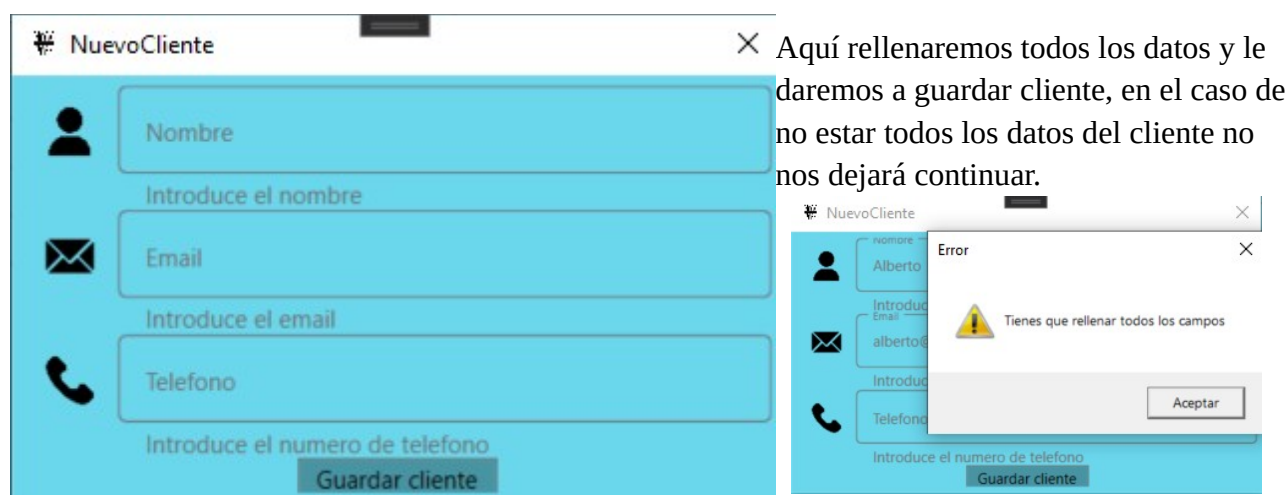
Haciendo clic en Clientes, nos llevará a la sección para gestionar los clientes.



Lo primero que veremos es una tabla con toda la información de nuestros clientes, si quisiésemos buscar, ordenar o filtrar por algún campo simplemente habría que darle a la flecha que hay al lado del campo que queremos usar como referencia y ahí podremos ordenar, filtrar y buscar como queramos.



Si queremos añadir un cliente, lo único que tendremos que hacer es hacer clic en el botón con un símbolo de un mas. Este nos abrirá una ventana con un formulario desde el cual se podrá crear un nuevo cliente.



Una vez hayamos rellenado todos los datos y le demos a guardar cliente, el nuevo cliente se guardará en la base de datos.

En el caso que queramos editar un cliente, tendremos que seleccionar el cliente que queremos editar haciendo clic encima de su fila y darle al botón de editar, que es el botón con un lápiz.

Magic Pabs App

# CLIENTES

idCliente	email	nombre	telefono
1	paco@gmail.com	Paco	689925147
2	mario@gmail.com	Mario	682120956
3	antonio@gmail.com	Antonio	634431201
5	carla@hotmail.com	Carla	652990132
6	lucia@gmail.com	Lucia	653337681
7	hugo@hotmail.com	Hugo	685772194
9	alberto@gmail.com	Alberto	604445337

+ ✎

Una vez le hayamos dado nos saldrá la misma ventana que para crear un cliente, pero esta vez, en vez de estar vacía estará con los datos del cliente que hayamos seleccionado.

NuevoCliente

nombre  
Hugo

Introduce el nombre

Email  
hugo@hotmail.com

Introduce el email

Telefono  
685772194

Introduce el numero de telefono

Guardar cliente

Ahora solo haría falta modificar los datos que queramos y darle al botón de guardar cliente, el cual modificará el cliente en la base de datos.

Importante recordar que si alguno de los campos esta vacío nos saltará un mensaje recordándonos que hay que rellenar todos los campos y no nos dejará guardar el cliente.

## 7.1.2) ESPECTÁCULOS

Para ir a la sección de espectáculos tendremos que hacer clic en la sección del menú que pone Espectáculos.

Magic Pabs App

# ESPECTACULOS

idEspectaculo	titulo	descripcion	duracion
1	Ahora	Espectaculo de magia familiar	30
2	Mentalismo	Espectaulo de mentalismo	30
3	Magia fabulosa	Un espectaculo de magia fabulos	45

+

Una vez estemos en la parte de espectáculos veremos que es bastante similar a la parte de Clientes, de hecho el funcionamiento es bastante similar también.

Nada mas acceder a esta parte, veremos toda la información importante de los espectáculos, como en el apartado de clientes también se podrá filtrar, ordenar y buscar por campos, para hacerlo, es igual que en el anterior, le damos a la flecha que hay al lado del campo por el queremos ordenar, filtrar o buscar y nos desplegará un menú con todas las opciones disponibles

Magic Pabs App

# ESPECTACULOS

idEspectaculo	titulo	descripcion	duracion
1	Ahora	Espectaculo de magia familiar	30
2	Mentalismo	Espectaulo de mentalismo	30
3	Magia fabulosa	Un espectaculo de magia fabulos	45

Sort A to Z

Sort Z to A

Clear Filter

Text Filters

Search

☒ (Select All)

☒ Ahora

☒ Magia fabulosa

☒ Mentalismo

OK

Cancel

+

Para crear un nuevo espectáculo tendremos que darle al botón con el signo de mas, nos abrirá una ventana con un formulario con el cual se podrá crear un nuevo espectáculo.

NuevoEspectaculo

Titulo

Introduce el titulo

Duracion

0

Introduce la duración del espectáculo en minutos

Descripcion

Describe como será el espectáculo

Guardar espectacu

Aquí rellenaremos todos los datos para poder crear un nuevo espectáculo.

Como en el formulario de cliente, si no rellenamos todos los campos nos saldrá un aviso de que faltan datos y no podremos guardar el espectáculo.

Una vez hayamos rellenado todos los datos, le daremos al botón de guardar espectáculo y nos guardará el nuevo espectáculo en la base de datos.

Para editar un espectáculo solamente tendremos que darle al la fila del espectáculo que queremos editar y luego darle al botón de editar, que es el que tiene un dibujo de un lápiz y nos aparecerá la misma ventana de crear un espectáculo solo que con los campos ya rellenos con la información del espectáculo seleccionado.

NuevoEspectaculo

Titulo

Magia fabulosa

Introduce el titulo

Duracion

45

Introduce la duración del espectáculo en minutos

Descripcion

Un espectaculo de magia fabulosa que dejara asombrado a todo aquel que dese verlo

Describe como será el espectáculo

Guardar espectacu

### 7.1.3) EVENTOS

Para ir a la parte de eventos, como siempre, hacemos clic en la parte que pone Eventos en el menú.

Magic Pabs App

# EVENTOS

idEvento	tipoEvento	direccion	fecha
1	cumpleaños	calle prueba 1	01/03/2022
2	Boda	calle prueba 2	12/07/2022
3	Presentación	calle la cerámica 24	31/05/2022

Ver Espectaculo Ver Cliente

Una vez estemos en la parte de eventos, vemos que aunque sea bastante similar a las anteriores ventanas, hay un par de funcionalidades nuevas.

Como siempre, si queremos filtrar ordenar o buscar, tendremos, como siempre, que darle a la flecha que se encuentra al lado del campo por el que queremos filtrar, ordenar o buscar.

Magic Pabs App

# EVENTOS

Sort A to Z

Sort Z to A

Clear Filter

Text Filters

Search

☒ (Select All)

☒ Boda

☒ cumpleaños

☒ Presentación

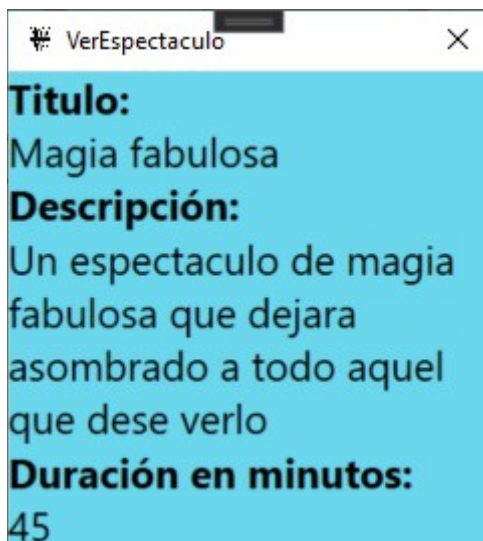
OK Cancel

direccion	fecha
calle prueba 1	01/03/2022
calle prueba 2	12/07/2022
calle la cerámica 24	31/05/2022

Ver Espectaculo Ver Cliente



Los eventos los contrata un cliente y tienen un espectáculo asociado, para ver toda esa información simplemente tendremos que darle al botón de Ver Espectáculo o Ver Cliente, esto nos abrirá una ventana mostrándonos toda la información.

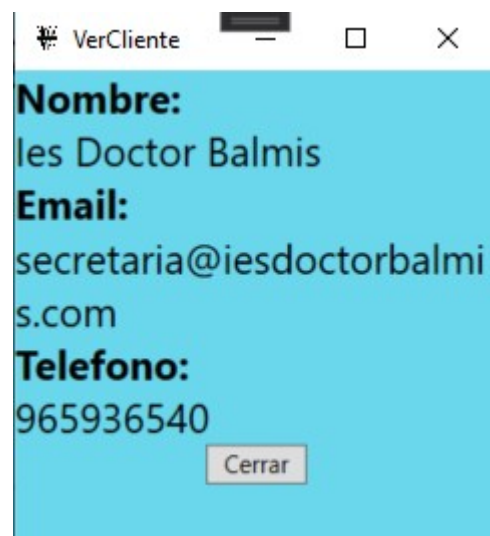


VerEspectaculo

**Título:**  
Magia fabulosa

**Descripción:**  
Un espectaculo de magia fabulosa que dejara asombrado a todo aquel que dese verlo

**Duración en minutos:**  
45



VerCliente

**Nombre:**  
Ies Doctor Balmis

**Email:**  
secretaria@iesdoctorbalmis.com

**Telefono:**  
965936540

Cerrar

Para crear un nuevo evento, lo haremos como siempre, hacemos clic en el botón de nuevo y nos aparecerá un formulario para crear un nuevo evento.



NuevoEvento

**Dirección**  
Introduce la dirección del evento

**Fecha**  
Introduce la fecha del evento (dd/mm/yyyy)

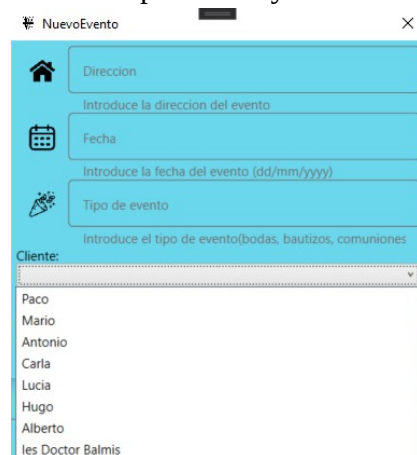
**Tipo de evento**  
Introduce el tipo de evento(bodas, bautizos, comuniones)

**Cliente:**

**Espectáculo:**

Guardar Evento

Este formulario, es un poco diferente a los anteriores, porque incluye dos combo-box desde los que se pueden seleccionar tanto un cliente ya creado como un espectáculo ya creado.



NuevoEvento

**Dirección**  
Introduce la dirección del evento

**Fecha**  
Introduce la fecha del evento (dd/mm/yyyy)

**Tipo de evento**  
Introduce el tipo de evento(bodas, bautizos, comuniones)

**Cliente:**

- Paco
- Mario
- Antonio
- Carla
- Lucia
- Hugo
- Alberto
- Ies Doctor Balmis

Con esto, podremos crear eventos asociándolos a clientes y espectáculos.

Como siempre si algún campo está vacío o el campo de fecha no se corresponde con el formato DD/MM/YYYY nos saldrá un mensaje avisándonos del error y no nos dejará crear el evento.

Una vez se haya rellenado todo, le daremos a guardar evento y el evento se guardará en la base de datos.

Por último si queremos editar un evento, solamente tendremos que seleccionar la fila correspondiente al evento que queramos editar y luego darle al botón de editar, entonces nos saldrá



el mismo formulario que para crear un nuevo evento, pero con los datos del evento seleccionado para que podamos editarlo.

NuevoEvento

 Direccion  
calle la cerámica 24

 Introduce la direccion del evento  
Fecha  
31/05/2022

 Introduce la fecha del evento (dd/mm/yyyy)  
Tipo de evento  
Presentación

Introduce el tipo de evento(bodas, bautizos, comuniones)

Cliente:  
les Doctor Balmis

Espectaculo:  
Magia fabulosa

Guardar Evento

✕ Modificaríamos los datos que queramos y le daríamos a guardar evento para que los cambios se guarden en la base de datos.

## 7.2) MANUAL DE LA APLICACIÓN DE ANDROID



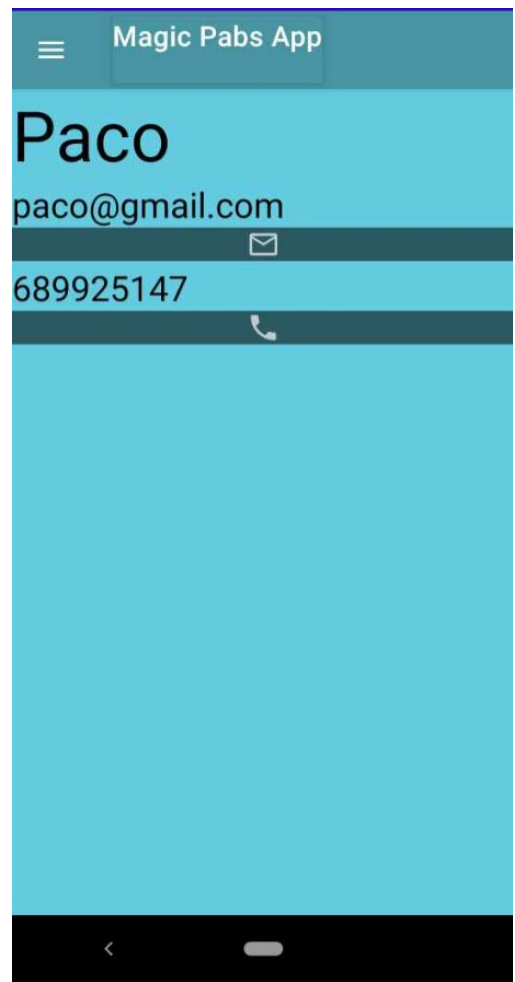
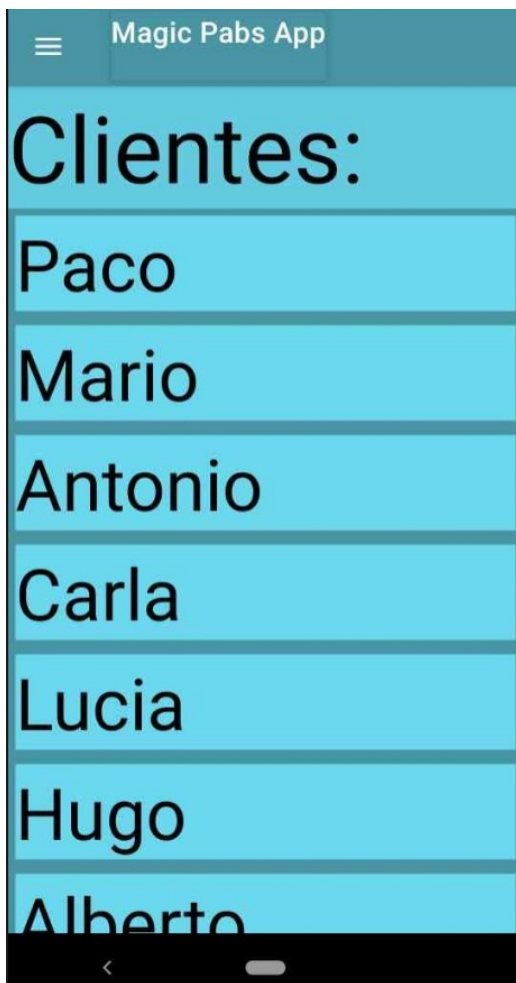
Nada mas entrar a la aplicación de Android nos encontraremos una pantalla de inicio muy similar a la de la aplicación de Windows.

La aplicación contiene un menú lateral desde el cual podremos navegar entre las diferentes pantallas de la aplicación.



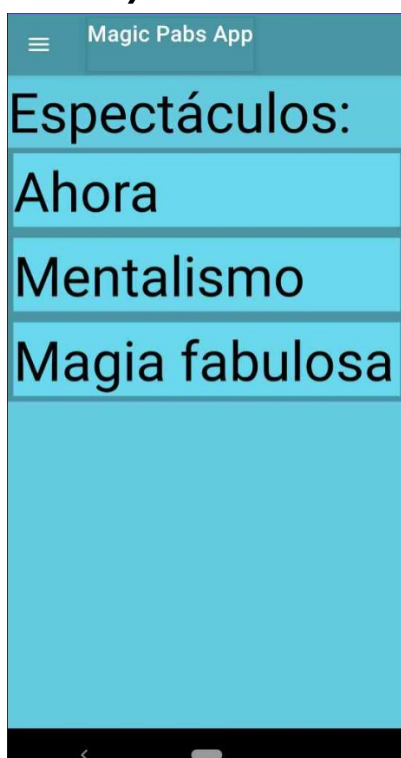
### 7.2.1) CLIENTES

Una vez hagamos clic en Clientes, nos redirigirá a la parte de clientes, la cual nos mostrará una lista con todos los nombres de todos los clientes, Si queremos ver la información de algún cliente, solamente tendremos que hacer clic encima de su nombre para que nos muestre toda la información.



Una vez en la pantalla con la información del cliente, tendremos dos botones, uno que llamará al numero del cliente, y otro que servirá para mandar un correo al cliente. Con estos dos botones se nos hará mucho mas fácil la comunicación con nuestros clientes.

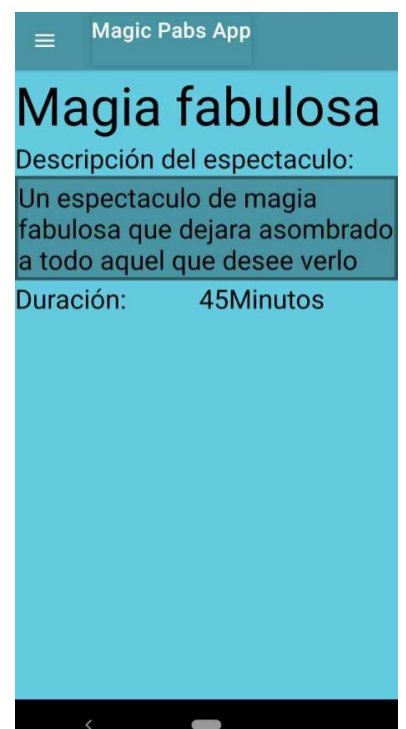
## 7.2.2) ESPECTÁCULOS



Para acceder a la parte de espectáculos, solamente tendremos que hacer clic en la parte del menú donde pone espectáculos y nos llevará a ver todos los espectáculos disponibles.

Para poder ver el espectáculo en mas profundidad hay que hacer clic en el nombre del espectáculo, el cual nos llevará a otra pantalla con toda la información del espectáculo.

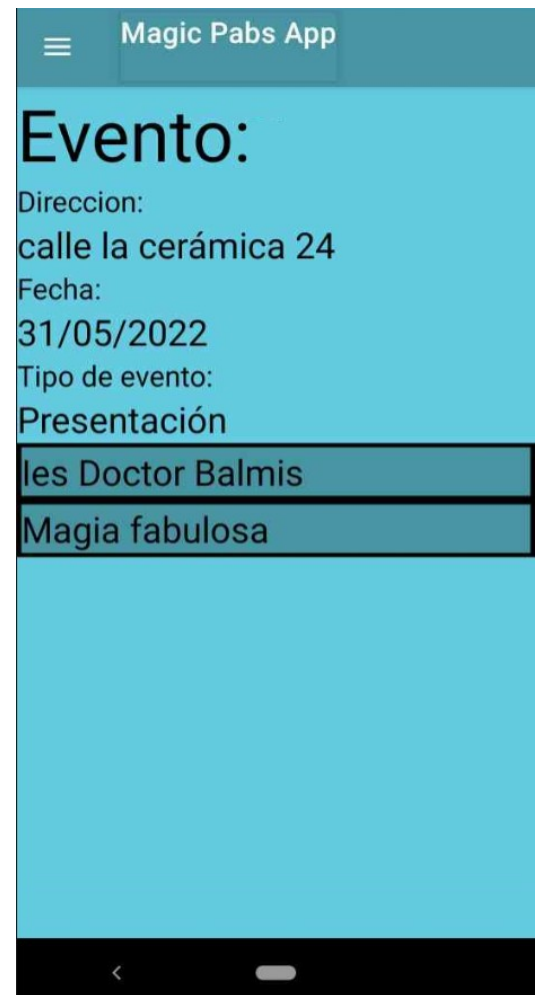
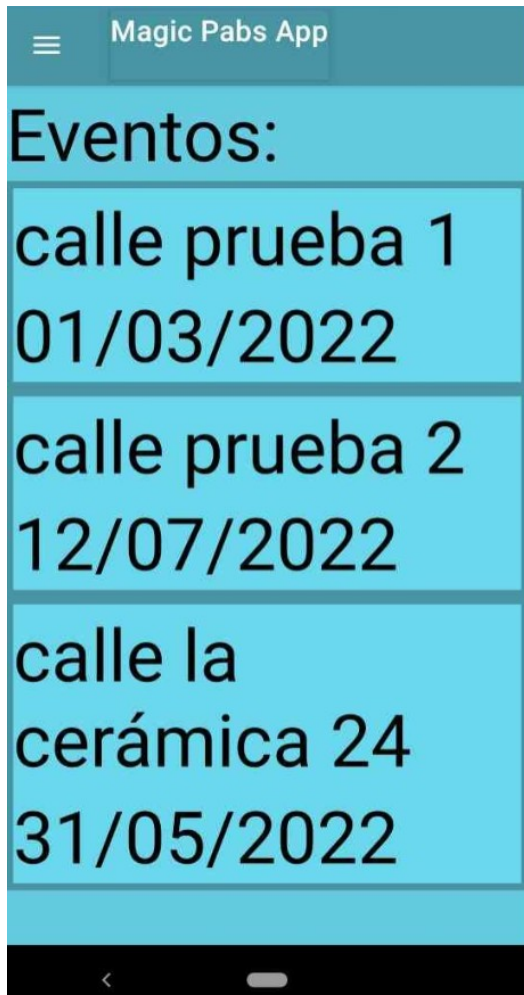
Desde esta pantalla podremos ver toda la información del espectáculo: el titulo del espectáculo en grande arriba del todo, en medio tenemos la descripción del espectáculo, y por último la duración.



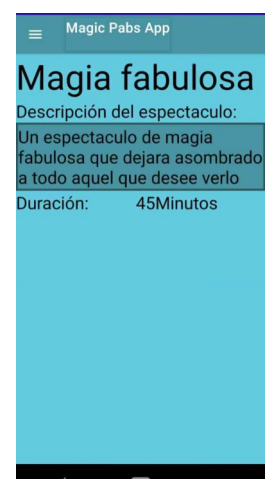
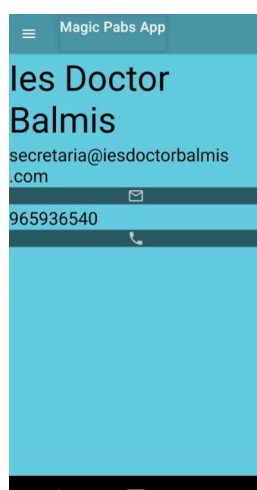
### 7.2.3) EVENTOS

Como siempre, para acceder a la pantalla de eventos haremos clic en donde pone eventos en el menú lateral.

Una vez estemos en la pantalla de eventos, nos saldrán todos los eventos mostrándonos la dirección del evento y su fecha. Para obtener toda la información del evento, simplemente haremos clic sobre el evento. Esto nos llevará a una pantalla con toda la información del evento.



Si queremos saber mas información sobre el cliente que ha contratado el evento o el espectáculo que se va a realizar en ese evento, tendremos que pulsar sobre el botón con el nombre del cliente o sobre el botón con el nombre del espectáculo. Esto nos llevará a una pantalla con toda la información del espectáculo o del evento.

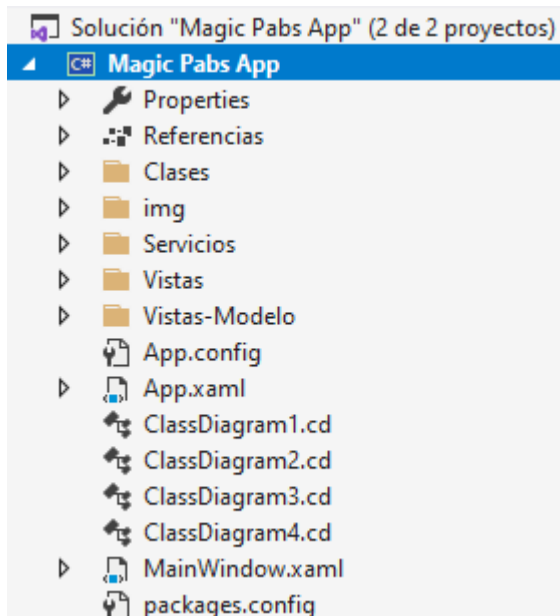


## 8) REQUISITOS E INSTALACIÓN

### 8.1) DESCRIPCIÓN DEL ENTREGABLE

#### 8.1.1) WINDOWS

La aplicación de Windows consta de los siguientes contenidos divididos en carpetas:



-**Clases:** En esta carpeta están todas las clases de datos.

-**img:** En esta carpeta estarán todas las imágenes que se usan en la aplicación.

-**Servicios:** En esta carpeta estarán todas las clases estáticas que hacen algún servicio, como conectarse con la API o navegar por las diferentes ventanas.

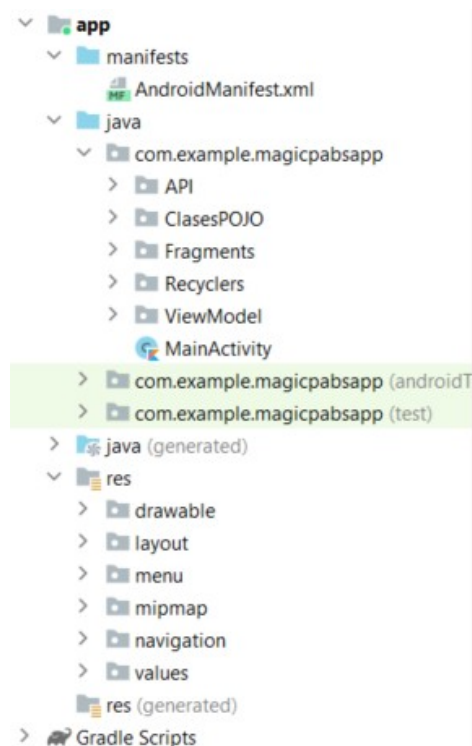
-**Vistas:** Esta es la carpeta que contiene las vistas de las ventanas y los diferentes user-controls.

-**Vistas-Modelo:** Esta carpeta tiene todas las vistas-modelo, el código que se encarga de gestionar las vistas.

A parte de todas las clases que están dentro de las carpetas descritas anteriormente, esta MainWindow.xaml que es el archivo con la vista principal de la aplicación.

#### 8.1.2) ANDROID

La aplicación de Android consta de los siguientes contenidos divididos en las siguientes carpetas:



-**manifests:** En esta carpeta se encuentra el archivo AndroidManifest.xml, el cual es un archivo de configuración sobre ciertos aspectos de la aplicación como el nombre, la actividad donde empieza la aplicación, los permisos...

-**java:** Dentro de esta carpeta estarán las carpetas con todo el código de la aplicación:

-**API:** Aquí estarán las clases para poder conectar nuestra aplicación con la API.

-**ClasesPOJO:** Las clases de datos de la aplicación.

-**Fragments:** Las clases para gestionar el funcionamiento de los fragments.

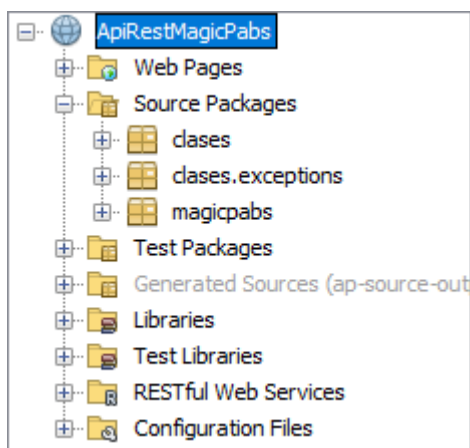
-**Recyclers:** Las clases con el código necesario para poder usar recycler-view en nuestras aplicaciones.

-**ViewModel:** Las clases para hacer posible el paso de datos entre fragments.

- MainActivity**: Es la clase con el código necesario para hacer funcionar la activity principal.
- res**: Dentro de esta carpeta se encuentran los archivos para todo lo relacionado con el apartado visual de la aplicación.
  - drawable**: Dentro de esta carpeta están todas las imágenes que usará nuestra aplicación.
  - layout**: Es la carpeta que contiene todo el diseño de todas las interfaces de la aplicación.
  - menu**: Es la carpeta que contiene los archivos de diseño de los menús que aparecen en nuestra aplicación.
  - mipmap**: Es la carpeta que contiene el icono de la aplicación en todas sus resoluciones.
  - navigation**: En esta carpeta está el archivo de configuración de la navegación de la aplicación.
  - values**: Esta carpeta contiene los archivos con las configuraciones de colores, temas, palabras...

### 8.1.3) API REST

La API REST tiene los siguientes contenidos divididos en las siguientes carpetas:



-**Web Pages**: Es la carpeta que contiene los archivos HTML, CSS que mostrará la página.

-**Source Packages**: Aquí se encuentra el código de la API

- clases**: Como su nombre indica, dentro de este paquete se encuentran todas las clases de datos que necesitará la API REST junto a los controladores que harán que se pueda conectar a la base de datos.

- clases.exceptions**: Un paquete con unas clases para gestionar algunas excepciones de las clases.

- magicpabs**: Aquí están las clases con el código necesario para que la API pueda proporcionar los datos de la base de datos en formato JSON.

-**Libraries**: En esta carpeta se encuentran las librerías que usa nuestra API.

-**RESTful Web Services**: es la carpeta con los archivos que atenderán los métodos del API REST.

### 8.2) REQUISITOS:

Para poder instalar Magic Pabs App necesitaremos:

- Un servidor linux con MySQL y Wildfly
- Un ordenador con Windows
- Un dispositivo móvil con Android.

## 8.3) INSTALACIÓN DE LA API REST EN UN SERVIDOR LINUX

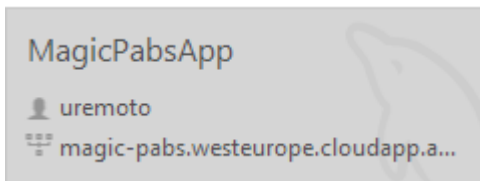
Lo primero que tendremos que hacer para instalar la API REST en nuestro servidor es, una vez creado nuestro servidor con Linux es instalar MySQL, para ello ejecutaremos el siguiente comando:

```
# sudo mysql -u root -p
```

Una vez hecho habra que crear un usuario de MySQL con los siguientes comandos:

```
mysql> CREATE USER 'uremoto'@'%' IDENTIFIED BY 'XXXXXXXX';  
mysql> GRANT ALL ON *.* TO 'uremoto'@'%';
```

Ahora accedemos desde nuestro MySQL Workbench al servicio de MySQL del servidor



Una vez hayamos accedido ejecutamos el script “Magic Pabs App.sql”

Ahora instalaremos Wildfly, para ello usaremos el siguiente comando:

```
# sudo wget https://github.com/wildfly/wildfly/releases/download/25.0.1.Final/  
wildfly-25.0.1.Final.tar.gz
```

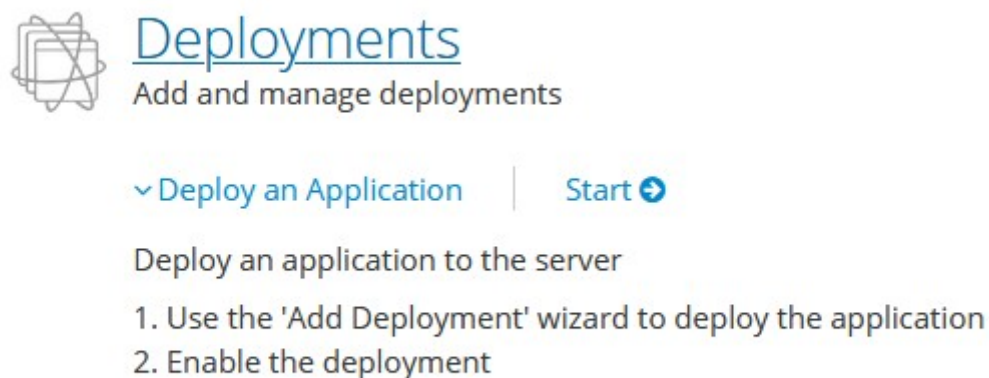
Y ahora ejecutaremos el archivo de instalación:

```
# sudo /opt/wildfly/bin/add-user.sh
```

Rellenamos la información que nos pide y ya tendríamos el servidor funcionando.

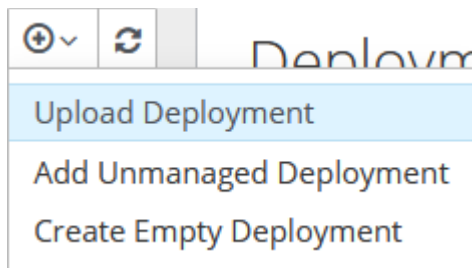
Ahora accederemos al puerto 8081/console de nuestro servidor para acceder al panel de control de Wildfly.

Una vez allí podremos hacer el deploy de nuestra API REST, para ello haremos clic donde pone start:



Una vez en la parte de deployments le daremos a un botón con un mas y luego a Upload Deployment.





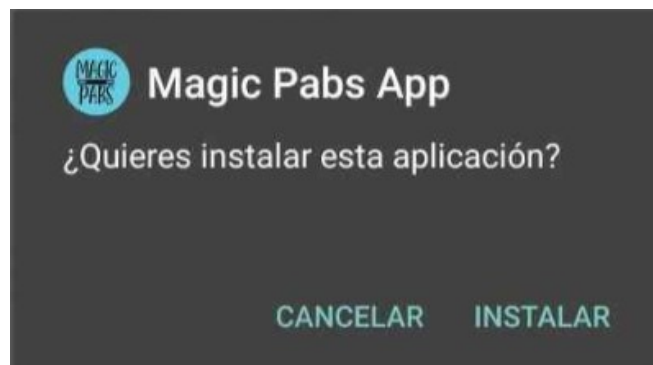
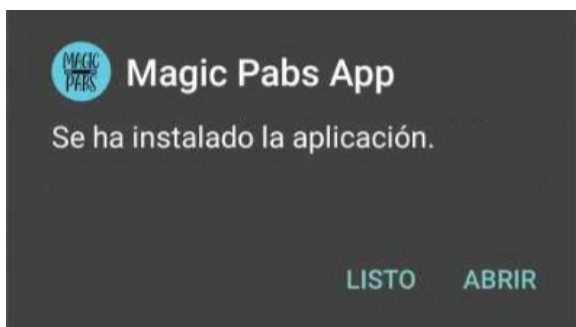
Ahora solamente habrá que arrastrar el archivo “ApiRestMagicPabs.war ” y ya tendríamos la API REST funcionando.

## 8.4) INSTALACIÓN DE LA APLICACIÓN DE ANDROID

Para instalar Magic Pabs App en un móvil con android solamente tendremos que ejecutar el archivo “Magic Pabs App.apk”

Una vez hayamos abierto el archivo, lo primero que nos saldrá es el siguiente mensaje en el cual tendremos que pulsar Instalar

Después de esto se instalará la aplicación en nuestro dispositivo y nos saldrá el siguiente mensaje.

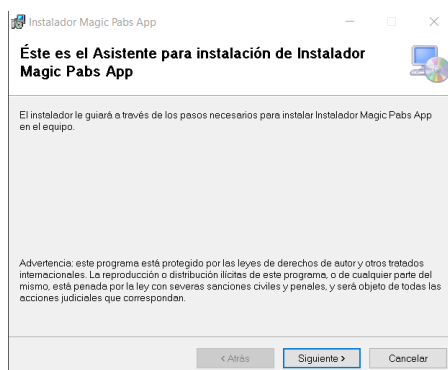


Una vez nos haya salido este mensaje la aplicación estará instalada en nuestro dispositivo, podremos darle a abrir si queremos abrirla en ese momento, o darle a listo y abrirla mas adelante.

## 8.5) INSTALACIÓN DE LA APLICACIÓN DE WINDOWS

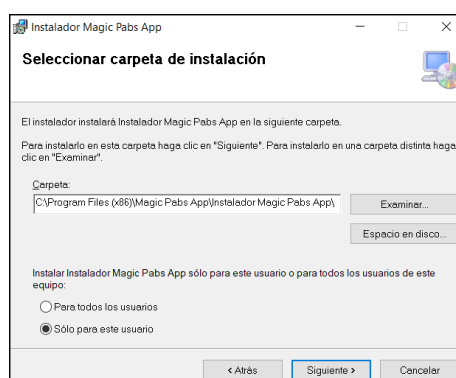
Para poder instalar Magic Pabs App en nuestro ordenador con Windows hay que ejecutar el instalador llamado “Instalador Magic Pabs App.msi”.

Nada mas ejecutarlo nos aparecerá la siguiente ventana.

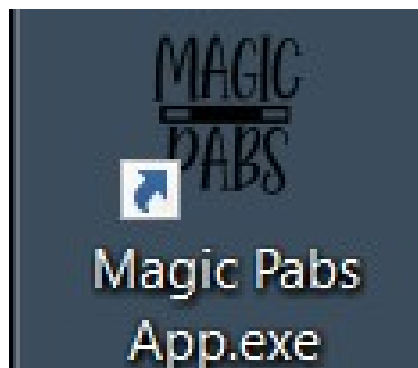
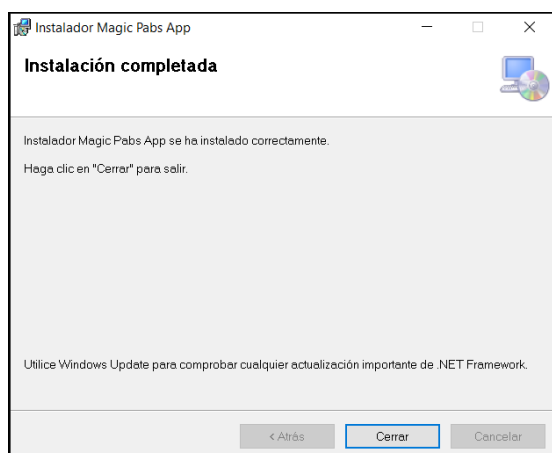


Para continuar con la instalación le daremos a siguiente.

En el siguiente paso de la instalación nos pedirá una ruta de donde se instalará la aplicación, podemos elegir nosotros una ruta en específico o dejar la ruta por defecto.



Continuaremos dándole a siguiente hasta que nos salga la ventana avisándonos de que la instalación ha sido completada. Ahora si ya tendremos instalada la aplicación en nuestro ordenador Windows.



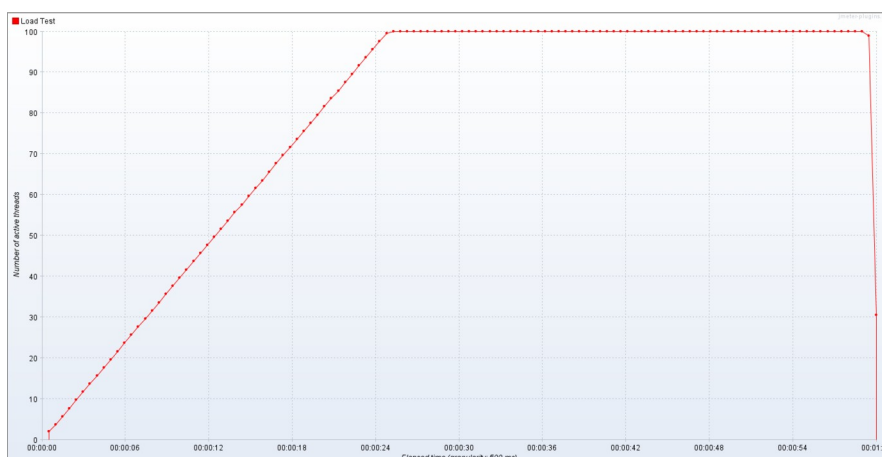
## 9) TESTING Y ANÁLISIS DE RENDIMIENTO

Para este proyecto he querido incorporar parte de los conocimientos que he adquirido durante mi periodo de FCT haciendo unos pequeños tests de rendimiento de la API REST, he decidido hacer el análisis de rendimiento con la herramienta de Jmeter, y he generado 4 pruebas de rendimiento:

- Test de carga. Consiste en aplicarle una carga de usuarios y peticiones con la que el servicio tendrá que trabajar sin problema.
- Test de escalabilidad. Consiste en ir aplicándole cada vez mas carga de usuarios y peticiones para ver a hasta que cantidad de peticiones se comporta de forma correcta.
- Test de estrés. Consiste en aplicarle una carga superior a la carga máxima con el objetivo de monitorizar el servicio con cargas para las cuales no esta preparado.
- Test de picos. Consiste en test de carga normales a los cuales en cierto punto de la ejecución se le insertará un pico de usuarios.

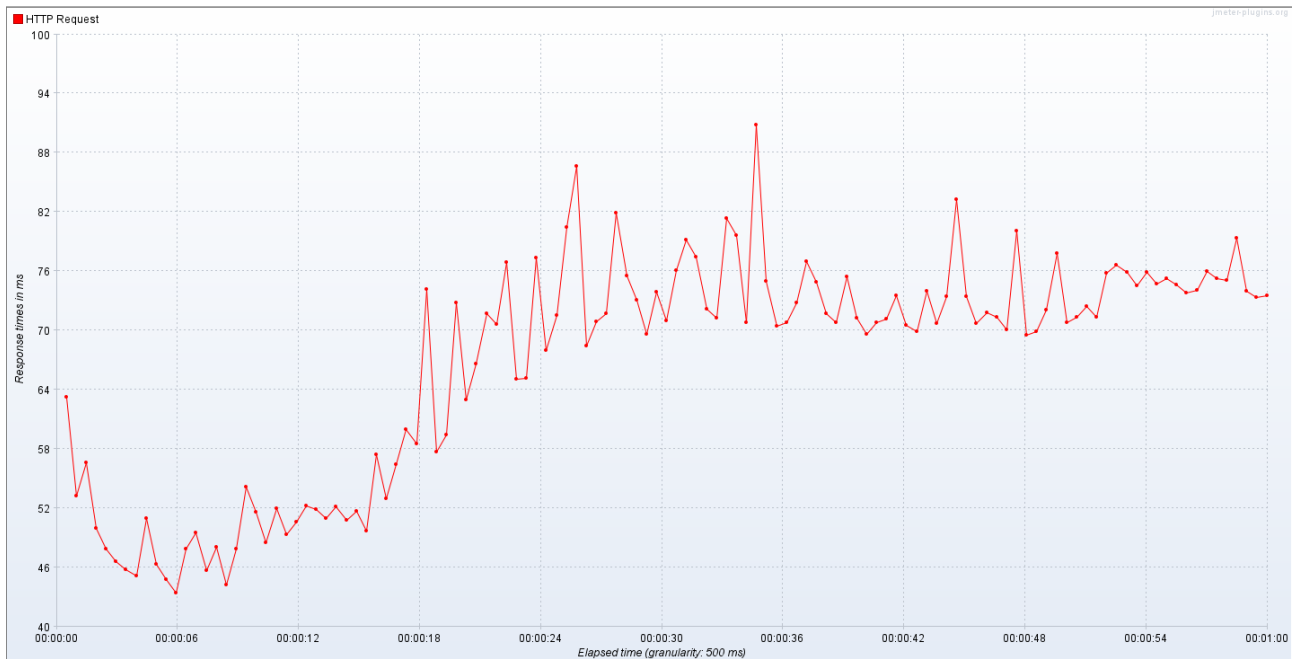
### 9.1) TEST DE CARGA

Para este test le metí una carga de 100 usuarios durante 1 minuto con una rampa de entrada de unos 25 segundos.

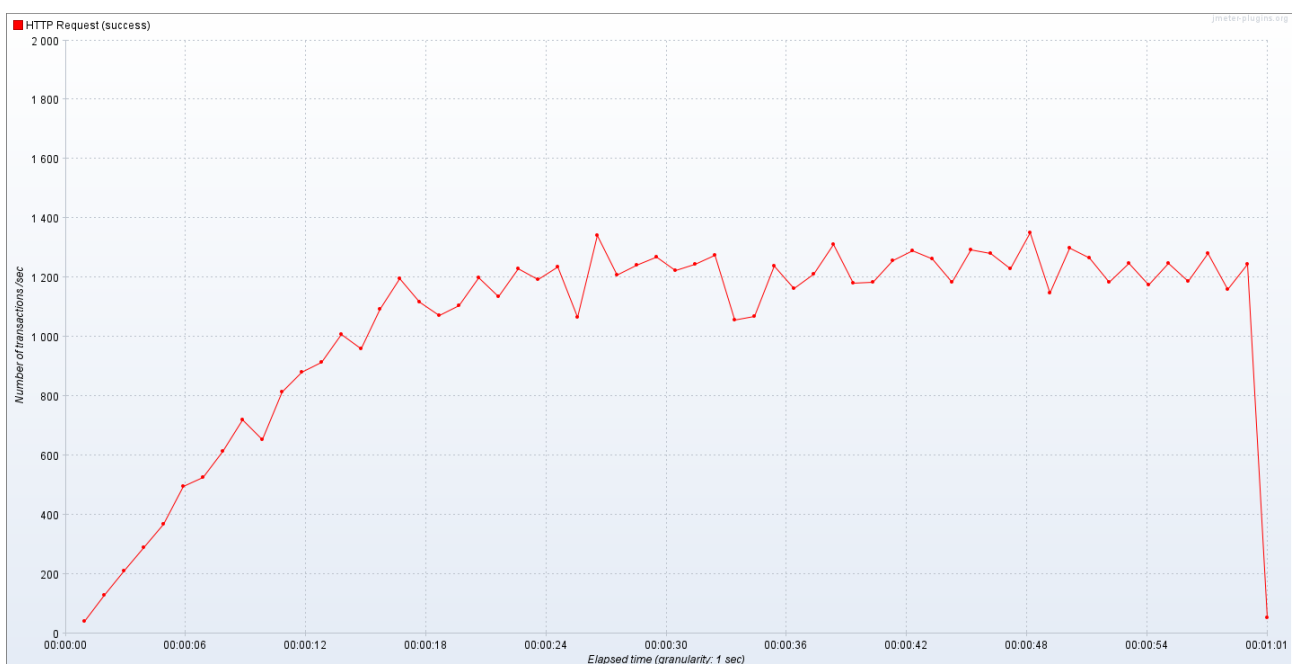




## Resultados del tiempo de respuesta:



## Resultados del numero de peticiones:

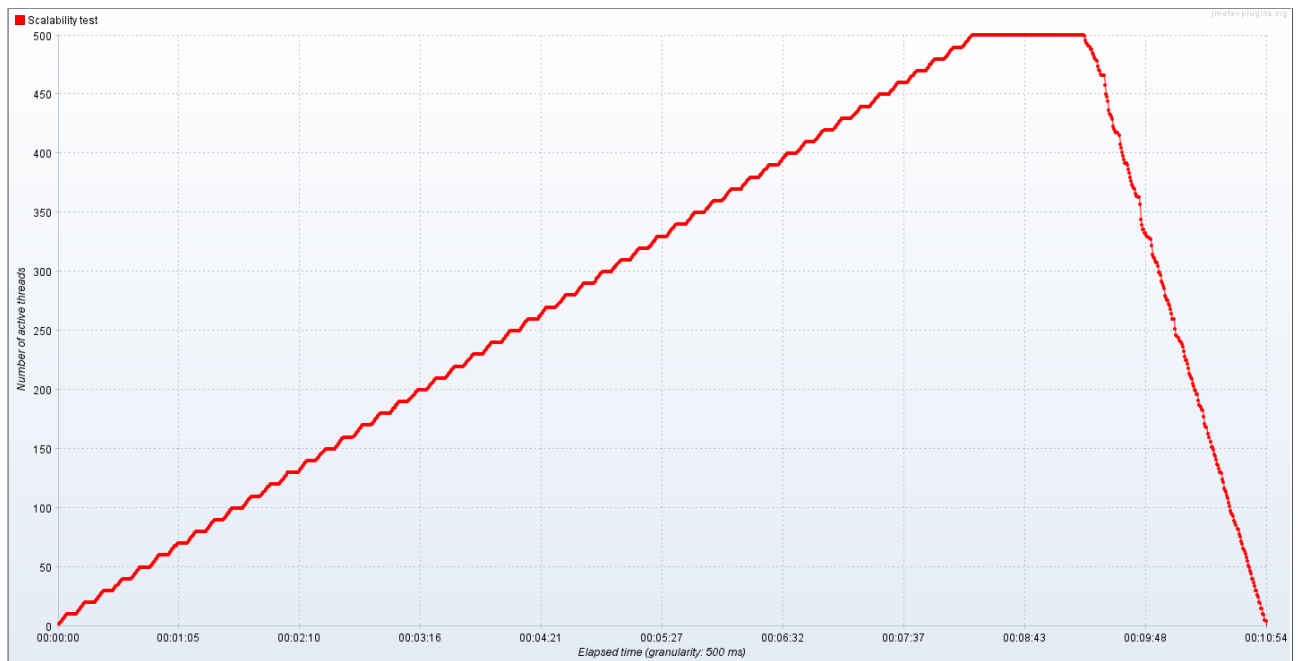


Se puede observar como al subir el numero de peticiones el tiempo de respuesta aumenta, pero aún aumentar y estar atendiendo a un gran numero de peticiones se mantiene en buenos tiempos de respuesta y bastante estables.

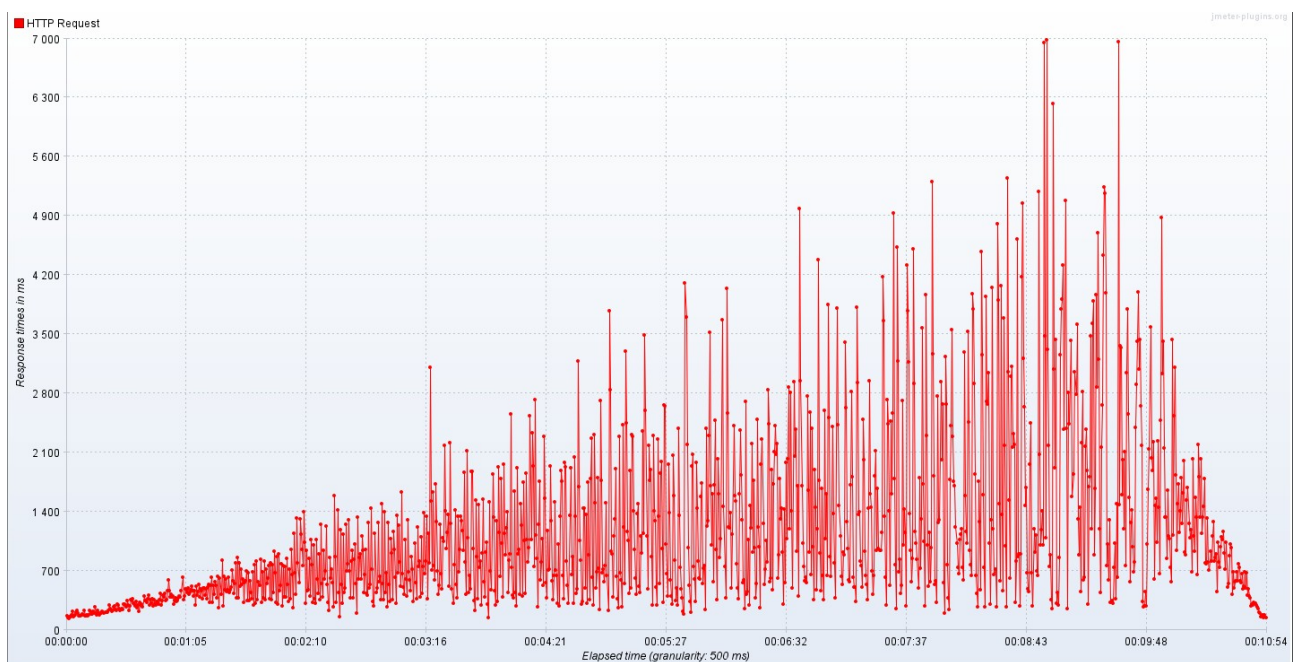
## 9.2) TEST DE ESCALABILIDAD

En este test se busca encontrar a partir de que cantidad de usuarios y peticiones el servicio deja de funcionar como debería.

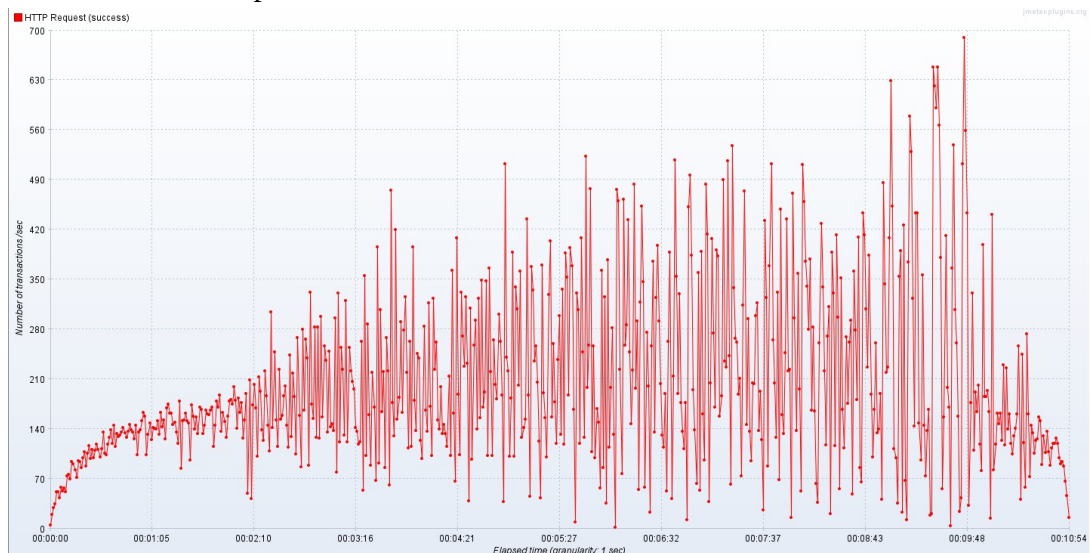
Para la prueba vamos a meter una carga de 500 usuarios con una rampa de 10 usuarios cada 5 segundos con una rampa de 5 segundos.



## Resultados del tiempo de respuesta:



## Resultados del numero de peticiones:

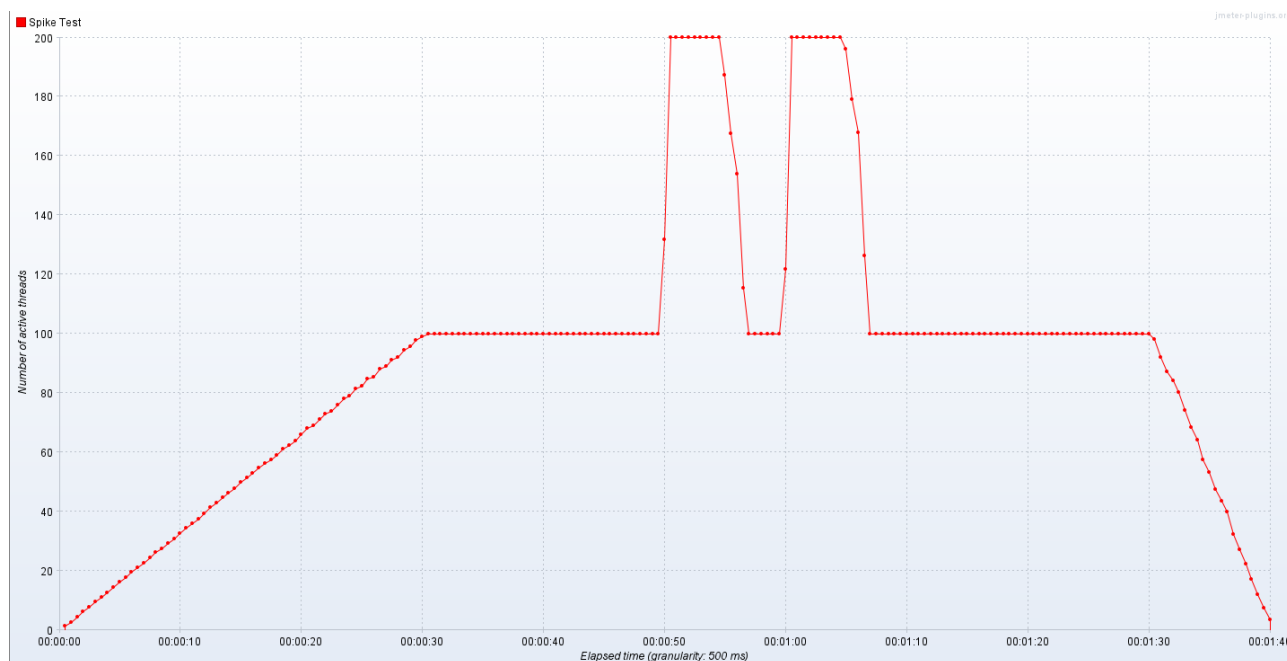


Se puede observar como al principio obtenemos buenos tiempos de respuesta, pero conforme vamos aumentando la carga de usuarios los tiempos de respuesta se disparan, sobre todo a partir de los 150-200 usuarios los tiempos de respuesta empiezan a sobrepasar los 1000ms con lo cual podríamos establecer en ese rango de usuarios el máximo número de usuarios simultáneos haciendo peticiones.

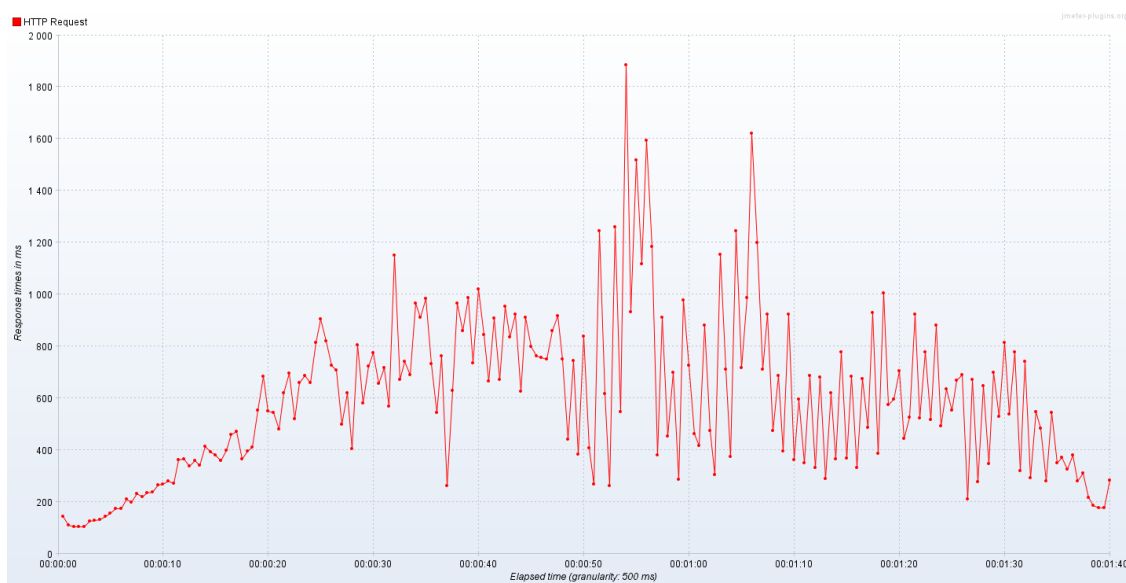
### 9.3) TEST DE PICOS

Como hemos comentado anteriormente, en este test buscamos incorporar picos de usuarios y peticiones para ver el comportamiento de este ante este tipo de situaciones.

Para este test le hemos metido una carga de 100 usuarios con una rampa de 30 segundos y dos picos de otros 100 usuarios durante 5 segundos cada uno con 5 segundos de tiempo de espera entre ellos.



Resultados del tiempo de respuesta:



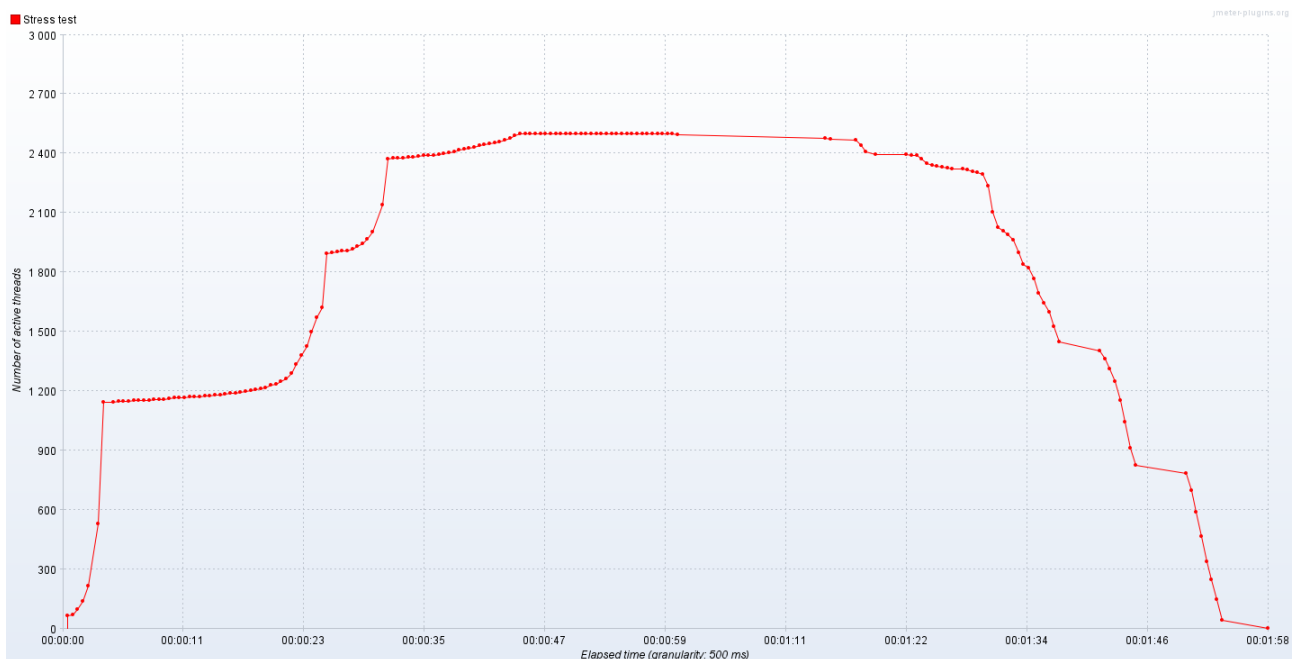
Resultados del número de peticiones:



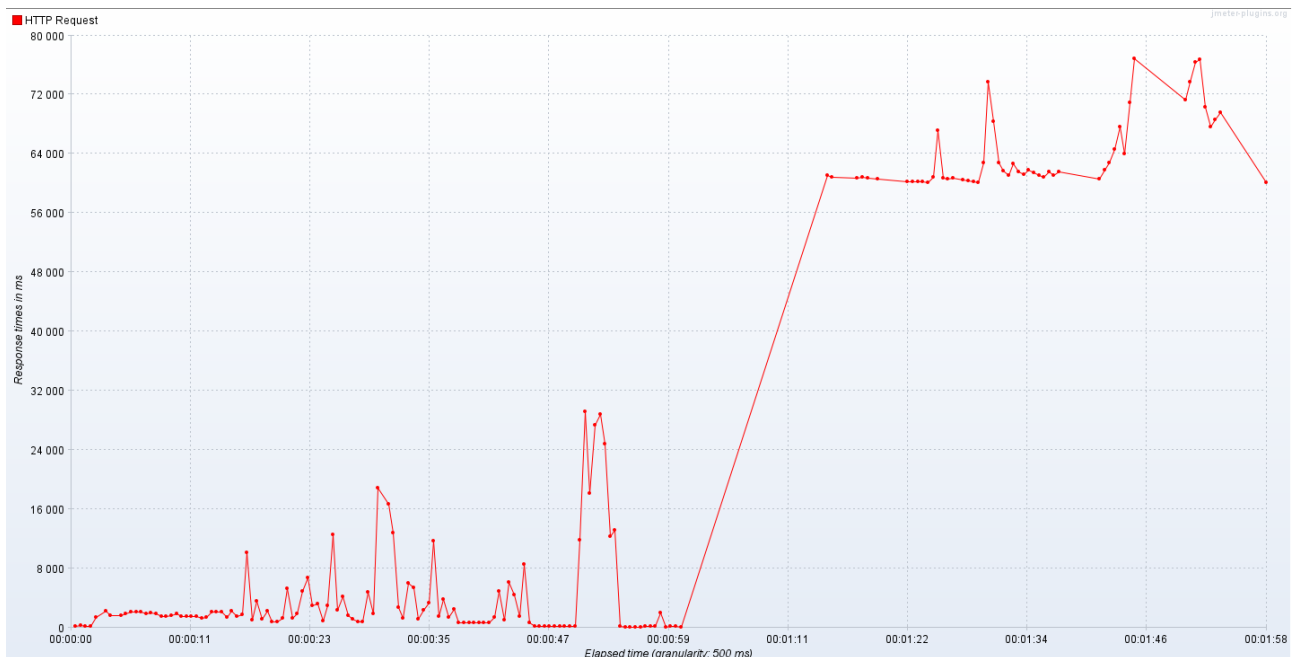
Se puede observar como con 100 usuarios se mantiene en tiempos de respuesta normales, aunque con algún pico por encima de los 800ms, para esa cantidad de usuarios y como al meterle esos picos de usuarios, el tiempo de respuesta crece bastante y genera picos de hasta 2000ms

## 9.4) TEST DE ESTRÉS

En este test queremos probar que tal se comporta nuestro servicio con una carga bastante superior a la carga con la que suele trabajar normalmente, para este caso decidí meterle una carga de 2500 usuarios.



Resultados del tiempo de respuesta:



Resultados del numero de peticiones:



Se puede observar como los tiempos de respuesta van subiendo conforme los usuarios van aumentando, y que una vez que los 2500 usuarios están haciendo peticiones los tiempos de respuesta empiezan a aumentar llegando incluso a los 78000ms e incluso las ultimas peticiones no llegan a tener una respuesta correcta.

## 10) CONCLUSIONES

### 10.1) CONCLUSIONES SOBRE EL TRABAJO REALIZADO

Este proyecto, ha sido un proyecto con el que he podido asentar todos los conocimientos de las diferentes asignaturas de este grado superior y ponerlas todas en común en un proyecto el cual

podiese acabar utilizando yo personalmente en la vida real, ya que la idea del proyecto nace de una necesidad personal.

Ha sido un proyecto con ciertas dificultades, ya que ha sido la primera vez que me ha tocado juntar todos los contenidos de las diferentes asignaturas y aunque muchas de las cosas que hay en el proyecto las habíamos realizado anteriormente en clase, al juntarlas entre ellas dan pie a errores y problemas con los que nunca me había encontrado, sumado a las cosas que he querido añadir al proyecto que no se habían dado en clase, he acabado con un proyecto, que no solo me ha servido para asentar conocimientos, sino que también me ha servido para aprender bastantes cosas nuevas.

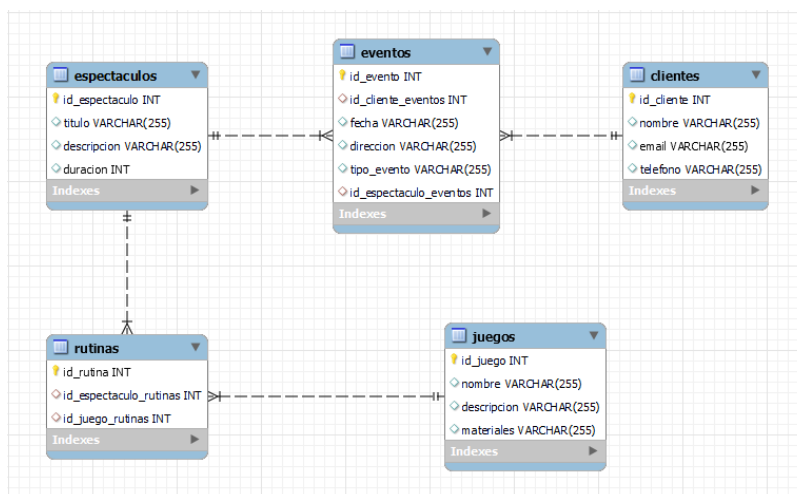
Para resumir, estoy muy contento y orgulloso del resultado final de este proyecto. He conseguido que a partir de una necesidad que tenía, poder crear una solución informática aplicando los conocimientos adquiridos durante estos dos años.

## 10.2) POSIBLES AMPLIACIONES Y MEJORAS

Una posible ampliación que me gustaría poder aplicar en un futuro sería la de tener una página web desde la cual los clientes pudiesen solicitar eventos a través de un formulario y que el ilusionista pueda aceptar o no el evento a través de las aplicaciones.

Otra posible ampliación que también me gustaría hacer en un futuro sería la de que a parte de toda la información con la que actualmente trabaja la aplicación como son los clientes, los espectáculos y los eventos, añadir que dentro de los espectáculos se pudiese almacenar la información de los juegos que se van a realizar en dicho espectáculo.

De hecho ya he empezado a trabajar en esta última posible ampliación y en modelo entidad relación quedaría tal que así:



## 11) BIBLIOGRAFÍA

-Apuntes de clase

-[Formaciones internas de Minsait](#)

-[StackOverflow](#)