Memoria del proyecto: Eventia

# Índice

1. Presentación del proyecto
   1. Presentación en español
   2. Presentación en inglés
2. Fases del proyecto

# Presentación del proyecto

## Presentación en español

# Eventia: Donde cada evento, importa

# 

Nuestro proyecto, Eventia, aborda la creciente necesidad de una herramienta eficiente y versátil para la gestión de eventos de eventos entre amigos. La necesidad de un sistema que facilite y optimice cada aspecto del proceso es más relevante que nunca. Eventia, busca ofrecer una solución integral y accesible para todos los organizadores de eventos.

La gestión de eventos tradicionales es un proceso complejo y laborioso. Los organizadores enfrentan desafíos como la coordinación de múltiples aspectos logísticos, la comunicación efectiva con los asistentes y la gestión de inscripciones. Muchos de estos procesos se manejan utilizando diversas herramientas que no siempre se integran de forma fluida. Esta situación plantea la necesidad de una herramienta unificada que centralice y simplifique la organización de eventos, asegurando una experiencia satisfactoria.

Para desarrollar Eventia, se ha adoptado una metodología ágil que permite iteraciones que coinciden con las tutorías de seguimiento del proyecto y feedback continuo. El proceso de desarrollo se ha dividido en cada una de esas iteraciones, enfocándose en la implementación de funcionalidades específicas. Esta metodología no solo facilita la detección y corrección temprana de errores, sino que también asegura que la plataforma evolucione de acuerdo a las necesidades específicas de la aplicación.

Eventia promete facilitar la forma en que se gestionan los eventos. Se espera que la aplicación reduzca el tiempo y los recursos necesarios para organizar un evento, mejorando al mismo tiempo la calidad de la experiencia para los asistentes.

En resumen, Eventia se posiciona como una herramienta esencial para cualquier persona que quiera agilizar la organización de eventos, con su enfoque integral y su capacidad de adaptarse a diferentes tipos de eventos. A medida que la tecnología continúa avanzando, herramientas como Eventia, facilitan la vida de los organizadores, mejorando su calidad de vida y el tiempo necesario para la creación de eventos.

## Presentación en inglés

Eventia: Where Every Event Matters

Our project, Eventia, addresses the growing need for an efficient and versatile tool for managing events among friends. In a world where event organization ranges from small gatherings to large international conferences, the need for a system that eases and optimizes every aspect of the process is needed more than ever. Eventia seeks to fill that gap by offering a comprehensive and accessible solution for all event organizers.

Traditional event management is often a complex and laborious process. Organizers face challenges such as coordinating multiple logistical aspects, effective communication with attendees and managing registrations. Many of these processes are handled using various tools that do not always integrate seamlessly, resulting in inefficiencies and errors. This situation creates a need for a unified tool that centralizes and simplifies event organization, ensuring a more coherent and professional experience.

For the development of Eventia, an agile methodology has been adopted, allowing iterations that match with project follow-up tutorials and a continuous feedback. The development process has been divided into these iterations, focusing on the implementation of specific functionalities. This methodology not only facilitates early detection and correction of errors but also ensures that the platform evolves according to the specific needs of the application.

Eventia promises to make easier the way events are managed. The application goal is to reduce the time and resources needed to organize an event while improving the quality of the experience for attendees.

In summary, Eventia positions itself as an essential tool for anyone looking to streamline event organization, with its comprehensive approach and ability to adapt to different types of events. As technology continues to advance, tools like Eventia not only make the lives of organizers easier, but also improve the quality and efficiency of event creation.

# Fases del proyecto

El primer paso de nuestro proyecto fue hacer una lluvia de ideas, cada uno de nosotros presentó sus propuestas a los demás detallando indicando qué aplicación se quería realizar y los puntos más importantes que debía incluir, así como su enfoque.

Una vez terminada, escogimos el tipo de aplicación, decidimos que sería una app para dispositivos móviles Android.

Después cuáles serían sus puntos claves:

* Una aplicación que ayudase a una gestión tanto laboral como personal de las actividades y eventos.
* Una aplicación en la que se pudieran incluir notas a los eventos, así como notas independientes.
* Una aplicación con una interfaz clara y sencilla
* Una aplicación que tuviera un gran alcance en rango de edad.
* Una aplicación que tuviera accesibilidad multilingüe para que se pudiera usar desde cualquier parte del mundo.

Así como aspectos técnicos:

* Usaremos el entorno de desarrollo Android Studio.
* Usaremos Java para desarrollar la aplicación.
* Usaremos Github para el control de versiones.
* La aplicación guardará los datos en una base de datos SQLite.
* Usaremos JUnit para realizar pruebas al código.

Con todos estos puntos definidos comenzamos instalando cada uno en nuestro entorno de trabajo las herramientas pertinentes:

* Instalamos la última versión del IDE Android Studio para ayudar a que no haya incompatibilidades durante el desarrollo.
* El IDE Android Studio ya viene con JUnit, por lo que no nos ha hecho falta instalarlo por separado
* SQLite ya viene por defecto para las aplicaciones Android y las bibliotecas vienen instaladas en el IDE Android Studio, por lo que no hemos tenido que instalarlo por separado.
* Instalamos Git en nuestros ordenadores.
* Creamos el proyecto en Android Studio y lo subimos a GitHub.
* Enlazamos nuestras cuentas de GitHub.
* Clonamos el proyecto en nuestro entorno de trabajo.
* Realizamos cambios menores para probar que todo funcionaba correctamente.
* Una vez realizadas todas las comprobaciones, vemos que todo funciona correctamente.
  1. Memoria provisional: este punto es para que vayamos escribiendo lo que vayamos haciendo, es un poco un como una caja de arena, vamos poniendo lo que vamos haciendo para luego organizarlo todos juntos. Mis comentarios en el Word los pondré en morado para que se vea que no pertenecen al texto final. Si os parece bien, os pongo una tabla a cada uno con un color asignado. Así podemos ir viendo que cambios necesitamos realizar, cosas que tengamos que comentar, partes del proyecto a destacar, etc. Gestión visual, más que nada.

Tabla de colores:

* + Vladyslav
  + Yosef
  + Antonio

Echadle un vistazo a lo que ya hay escrito hasta ahora, mirad que os parece y cualquier cosa la comentamos por aquí o por wasap =)

02/10/2024 (Yosef): Creo las activities de setting y splashscreen. Creo sus documentos de layout, modifico para darle aspecto. Creo en la activity main aspecto gráfico, incluyo widgets, funcionalidades a los menús y una toolbar funcional que. En setting creo una listview para poder cargar todas las configuraciones, según las vayamos teniendo. Creo también los ficheros de internacionalización y los ficheros de menú para darle aspecto a estos. Creo una animación para splashscreen y hago push.

03/10/2024 (Yosef): He implementado muchos cambios gráficos, sobre todo al main. He creado tres fragments para que sean los principales de la app. Uno de ellos tiene ya bastante funcionalidad que es el del calendario. Mi idea es poner el calendario en medio, el fragment today para que salgan los eventos que tenga el usuario en ese día a la izquierda y las notas genéricas a la derecha. El calendario del fragment calendario le he implementado un método para que devuelva el día que se está pickeando y un toast comentado para que podamos ir probando. En un futuro, mi idea sería que cuando se pincha un día en concreto se abriese una activity nueva con las notas, los eventos que tiene ese día el usuario y la pestaña de hoy que fuese un desglose del día actual pero más detallado, implementando una línea temporal como a las 12 reunion a las 14 comida y a las 16 gimnasio. Idme diciendo que os parece. El próximo día pondré todos los ‘botones’ con las imágenes escaladas, hay con color y sin color para dar un feedback al usuario indicando en cual está en todo momento, cuando pinchas uno se pone a color y cuando lo deseleccionas le ponemos el sin color.

06/10/2024 (Yosef): Los botones ya cambian de color al ser pulsados. Añado varios cambios al main y al fragment today para que se muestre el día actual, independientemente del idioma, un saludo y un listview que se llena con datos. El array que llena el listview se coge desde la activity main, pásandolo a través de un bundle. Realizo otros cambios como hacer que el listview se refresque deslizando hacia arriba del listview y cambios en el fichero gradle para añadirle una dependencia.

08/10/2024 (Vladyslav): Ha creado un nuevo layout que se carga tras seleccionar un día en el calendario. Ha incluido todas las strings necesarias para darle funcionalidad, ha dado funcionalidad al layout y actualizado los comentarios necesarios para esa parte. También, ha creado una rama para esta funcionalidad.

09/10/2024 (Yosef): Añado algunas funcionalidades al main y a las notas, solo pequeños cambios para poder cargar elementos.

11/10/2024 (Yosef): Creo una nueva rama de feature para las notas. Creo también una lista actualizable para ir cargando las notas y un botón flotante para ir añadiéndolas y le doy a todo funcionalidad, además de tener listos los comentarios de esta nueva parte y añadir strings.

16/10/2024 (Yosef): Creo un nuevo fragment para el sistema de amigos. Es mejor que no hagamos un drawer navigation ya que está siendo deprecados, es mejor que sigamos haciendo pestañas inferiores que es en lo que se enfocan ahora las aplicaciones Android. Échale un vistazo a las nuevas aplicaciones hechas por Google. No hay frameworks específicos para Android, hay recursos ya creados, pero son para dar cierta imagen, de momento los seguiré creando a mano, ya que queda mejor, más personalizado y sin fallos de compatibilidad. También resizeo todas las imágenes para que queden mejor y encajen bien en la parte inferior de pantalla, además de incluir nuevos iconos.

17/10/2024 (Yosef): Creo activities, xmls, lógica de programación y todo el aspecto visual para el login y el registro de usuarios en la aplicación. Solo resta conectarlo a firebase. También modifico todas las activities para que se muestren en vertical y no se puedan girar, rollo Instagram, que no deja que los contenidos se muestren en landing page por diseño de la propia aplicación. Añado muchas cosillas como strings, fuentes, backgrouns personalizados, personalziación de botones, etc. También creo nuevas clases para representar una nota y hacer se muestre un tipo de lista llamada recyclerview. Como viewholder o adapter.

19/10/2024 (Yosef): Creo la Firebase para la app. linkeo la app con Firebase, creo toda la lógica de programación para que un usuario pueda registrarse y también la parte de que un usuario se registre en una base para que pueda tener amigos. Creo varios usuarios de pruebas y hago varias pruebas de inicio. Creo también json para la base, modifico el json de Firebase, creo clase usuario para poder guardarlo en la app. También creo toda la lógica de Firebase necesaria en la propia base. Creo string y linkeo todas las necesidades de estas clases.

20/10/2024 (Yosef): Corrijo pequeñas cosas para poder mejorar las funciones de la app. Añado todas las mejoras necesarias a las notas para que estas sean exclusivas para cada usuario dependiendo de la base de datos. También actualizo los datos e incluyo algunas strings para que se muestren al usuario en su idioma.

22/10/2024 (Yosef): Creo el sistema básico de solicitudes de amistad. Los layouts y clases que creo vamos a necesitar, a demás de darle un aspecto chulo.

23/10/2024 (Yosef): Realizo el sistema para aceptar amigos que tenga un usuario en solicitudes de amistad. Linkeo con base de datos y esa parte es funcional completamente, falta por implementar el rechazo de solicitud, mostrar los amigos actuales en el FriendsFragment y función al botón de nueva solicitud de amigos

24/10/2024 (Yosef): Realizo el sistema para que salgan los amigos en Friendssearcher pero no funciona completamente, tiene que haber algún error cuando se cogen los datos de firebase. De todas maneras, el resto de elementos son fucionales.

25/10/2024 (Yosef): Arreglo el sistema de amigos. Arreglo el sistema de que cuando buscas un amigo no puedas enviarle una notificación si es a ti mismo. Realizo cambios y hago que un amigo y un usuario sea la misma clase, es decir, lo que diferencia a un amigo de un usuario ahora es que esté anidado en el json, no los datos de la clase, menos redundancia, menos probabilidad de fallo y mayor consistencia de los datos. Hago también arreglos visuales para que esté todo más bonito y también realizo cambios en el signup para que se registren los usuarios mejor. Solo falta por implementar que los usuarios tengan una forma de rechazar las peticiones de amistad, pero después de todo esto parece hasta sencillo. La forma de hacerlo será que en friend\_requests del json se eliminen los datos de la petición y se elimine por lo tanto con notify la petición.

27/10/2024 (Yosef): Añado el método para rechazar peticiones de amistad. Solo faltan comentarios para algunas clases, los iré añadiendo según tenga tiempo. Cosas a implementar que creo necesarias e iré haciendo poco a poco.

* + - * Primero: que los usuarios tengan una foto de perfil y puedan subir una
      * Segundo: una activity que muestre los datos de cada usuario y los eventos a los que está unido y coincidan con los que **tiene** el usuario
      * Tercero: una forma de eliminar amigos
      * Cuarto: añadir al login un “recuérdame” para que un usuario no tenga que logearse varias veces.
      * Quinto: poner al usuario en la appbar y que pueda hacer logout
      * Sexto: añadir la appbar a todas las activities
      * Séptimo: unificar los estilos de todas las activities

No hay una preferencia como tal en este orden, pero son las cosas que me faltarían de momento por implementar. Creo que deberíamos centrar esfuerzos en los eventos para ya poder ir concretando al app.

--IMPORTANTE—cuando tenga los comentarios de las clases subiré todo lo que tengo hasta ahora al main para que tengamos todas las clases como la de usuario disponibles todos.

29/10/2024 (Yosef): Añado unas pocas funcionalidades como que se mantengan las nuevas clases en vertical también o que quito imports que finalmente no usé, también agrego todos los comentarios que estaban pendientes. Voy a hacer un Roadmap con lista de tareas en Github en previsión de la llegada del nuevo integrante del grupo, así nos será mucho más fácil ir haciendo un seguimiento de las tareas. Le asigno también un color para los comentarios que quiera ir realizando. Hago merge de la rama principal y esta para actualizar la rama main a su estado más actualizado. –Segunda anotación— He realizado la eliminación de las notas con un swipe, al final ha sido más fácil de lo que pensaba, al final el método de borrado ya lo tenía. He creado todas las tareas realizadas, las pendientes y me falta asignarlas cuando entre el nuevo integrante a vosotros, según os una al proyecto. Tendréis roles para hacer y deshacer lo que queráis y añadir y quitar tareas cuando mejor lo veáis.

30/10/2024 (Yosef): Añado a Antonio a la Firebase y al Github. Realizo cambios importantes ya que he hecho unos cambios en gradle, muchos cambios visuales para unificar el estilo de la app, hago pequeños cambios en algunas clases para mejorar su funcionalidad. Pongo también la app bar en todas las actividades secundarias, hago que se pueda ver la imagen del usuario y que se pueda subir una nueva, creando un cloud storage en Firebase y borro los ficheros que ya no voy a usar. Por todos estos motivos, hago push al main, aún a falta de comentarios en la clase de UserSettings, que es en la que se muestran los datos del usuario y le permite cambiar la imagen y algunas mejoras visuales que tiene esa clase todavía pendiente. Mirad que os parece el nuevo aspecto visual y funcionalidades y lo comentamos. La base de datos en la nube yo la veo importante para poder subir imágenes de los eventos, por ello tmb os subo al main los cambios pq así podemos disponer todos de ello.

31/10/2024 (Yosef): Añado la app bar a todas las actividades y le hago cambios para que pueda mostrar el nombre. Hago también cambios visuales a las notas y yo creo que ya están terminadas, no creo que necesite más cambios, tiene un aspecto visual muy correcto y se crean y destruyen muy rápido como nos pidió el profesor. Echadle un ojo y decidme que os parece. He añadido también que cuando se cambie la foto, se registre como una url en la base de datos para que se puedan recuperar y mostrar en las actividades y fragments que lo necesitan. De esta manera, todos los usuarios pueden tener imagen de perfil y puede mostrarse en la app. También he unificado diseños para que todo esté más bonito como nos pidió el profesor. He cambiado el aspecto visual de los usersettings para que se integre con el diseño de la app. También he retocado el tamaño de las imágenes para que la app no tenga que cargar tanto, ahora que todos los usuarios tienen imágenes. He actualizado el roadmap también. De la tarea del usuario en la appbar solo me falta que pueda hacer logout y de la tarea de los datos de cada amigo me queda hacer algo que sea resultón para mostrar los amigos y que se puedan eliminar. Dadle prioridad a terminar lo de los eventos ya que sería bueno que en la activity de los amigos se pueda ver en cuales coincide contigo.

03/11/2024 (Yosef): Realizo la función propuesta por Antonio de que al dar atrás en el sign up te devuelva al login.

04/11/2024 (Yosef): Realizo que se puedan borrar amigos de la app y que se pueda hacer logout del usuario de la app. También genero activities adaptadores y demás clases necesarias para el funcionamiento. Una vez dentro de un amigo para cambiar entre su información me ha parecido que poner una paginación de dos ventanas en plan twitter me ha parecido guay, echadle un ojo cuando podáis. Linkeo la app con FirebaseCli para poder usar cloudmessaging.

05/11/2024 (Yosef): Realizo todos los comentarios y pequeños cambios de código para mejorar la funcionalidad de las nuevas clases y actividades generadas. Realizo varios cambios visuales para que quedo todo más estético y acorde con el resto de la app, también se pueden ver los eventos a los que estáis inscritos tu amigo y tú, mostrando los datos más importantes del evento en un solo vistazo, con un estilo de “cajitas” donde en cada caja va un evento. Creo que ha quedado vistoso, echadle un ojo y me decís que os parece. Actualizo github indicando que ya se pueden ver los datos de los usuarios, por lo que completo esa tarea y hago un merge de esta rama con la main parcialmente para que estos cambios se hagan permanentes y podamos centrar completamente en hacer un evento y que se mande un mensaje al usuario. Hay algunos ficheros que pertenecen a functions de firebase, de momento no hacen anda porque no las he programado, pero al hacer merge no me deja quitarlas, de todas maneras, si finalmente no usamos cloudmessaging y usamos otra solución los podemos dejar por si queremos implementar otras cosas con FirebaseCli.

06/11/2024 (Yosef): Consigo un calendario personalizado en el que se puede poner un icono por cada evento registrado. Lo uso para la aplicación y hago varios cambios en el calendar fragment. Pongo arriba el calendario, y abajo se van a mostrar los próximos eventos que va a tener el usuario. Aún falta por implementar esa función. Realizo también una función básica y la parte visual de la creación de un evento.

07/11/2024 (Yosef): Realizao la función completa de crear un evento, que este se registe en la base de datos y que se pueda subir una foto de evento y que se registre como primer usuario el mismo usuario que crea el evento. Lo comento todo y hago algunos cambios que mejoran el aspecto visual y la parte programática de esta nueva actividad. Ahora que se registran los eventos en la base de datos sería interesante que Antonio se ponga a mirar como mostrar los eventos que tenga hoy el usuario en la pestaña hoy.

08/11/2024 (Yosef): Ahora se muestran todos los eventos del usuario con un icono en el calendario personalizado y todos los eventos futuros que tiene también, hago correcciones estéticas, hago cambios en alguna funcionalidad para que sea más eficiente y actualizo comentarios. Ya se pueden crear eventos y ver los que tienes junto a los futuros eventos en la misma pantalla.

09/11/2024 (Yosef): Realizo el login recuérdame. Pongo un toggle button con un background personalizado, cambio el inicio firebaseauth al splash, que es desde donde se inicia automáticamente el inicio de sesión ahora y cuando se deslogea debe volver a introducir datos dando al recuérdame para que se le queden registrados y volverá tener el inicio de sesión automático. También realizo varios cambios estéticos para mejorar el aspecto visual de la app y los algunos cambios menores para mejorar la visualización como que los eventos próximos se muestren en orden por fecha.

10/11/2024 (Yosef): Realizo el login recuerdame y varios cambios programáticos y estéticos, tanto en la splashscreen como en la página de login de lo que estuve haciendo el viernes, como que se carguen los eventos por fecha o cambiar algunos colores. Así como actualizar la memoria y el roadmap. Como llevo varios cambios muchos cambios importante hago merge también con el main.

11/11/2024 (Yosef): Realizo EventViewerActivity que sirve para visualizar eventos, en ella pongo dos botones que llevan a una pantalla para invitar a tus amigos y otra para aceptar o rechazar las invitaciones. Falta comentar la del viewer pero funciona completamente, de momento solo lo e integrado apra recyclerview que está debajo del calendario, pero lo pondré para más sitios. Cuando concrete más escribiré log en la memoria

12/11/2024 (Yosef): Ahora ya se pueden invitar a los amigos a un evento pulsando un botón, mañana intentaré terminar la logica de todo el proceso haciendo que puedan aceptar la petición de evento, pero de momento ya se pueden enviar peticiones de evento

13/11/2024 (Yosef): Hago la logica para todo el sistema de acptar y rechazar eventos. Ya se puede tener un amigo al que invitas a un evento, este lo puede aceptar o rechazar si lo acepta se une al evento. Puedes ver los eventos en los que coincides con otras personas que no son tus amigos pero no los puedes elimnar, puedes ver tus propios eventos si pinchas en ti mismo. He añadido que al dar atrás en la pagina principal se recargue de nuevo el calendar fragment para que se actualicen los nuevos eventos y el calendario. Mañana haré todos los comentarios de las nuevas clases y modificaciones, añadiré los strings y actualizaré el log.

14/11/2024: hago arreglos menores en varias clases. Hago varios logs para varias clases. Comento todas las clases, todo el código y actualizo el log. Mañana intentaré hacer el botón de eliminar evento y la manera de editar un evento, que va a ser compartida entre todos los amigos, a no ser que me digáis lo contrario. Ya que la aplicación es más personal que profesional. No creo que necesite una estructura de edición como un calendario laboral.

15/11/2024: Realizo la forma de salirse de un evento, además, si es el último usuario incluido en ese evento, este se elimina. También realizo unos cambios visuales en EventViewerActivity para incluir los nuevos botones.

16/11/2024: Ahora los eventos se actualizan tras crear uno o al eliminarse y los eventos de nuestros amigos y propios se muestran por orden en el FriendViewer. Creo también varios paquetes para ordenar todas las clases creadas y comento las nuevas mejoras que he implementado. Como he creado muchas funciones importantes y el profesor me indicó pornerme con la memoria, subo todos estos cambios haciendo merge con el main, para que los tengáis disponibles, ya que son importantes para la lógica de funcionamiento del programa. También actualizo log.