Memoria del proyecto: Eventia

# Índice

1. Presentación del proyecto
2. Fases del proyecto

# Presentación del proyecto

En el entorno actual, los usuarios de Android enfrentan un problema común: la falta de una aplicación completa que les permita gestionar notas y planificar actividades y eventos de manera eficiente. Actualmente, muchas personas se ven obligadas a recurrir a múltiples aplicaciones para cumplir con estas funciones, lo que complica el seguimiento de tareas y afecta la productividad.

Nuestro objetivo es resolver esta problemática mediante el desarrollo de Eventia, una aplicación que unifica todas las funciones de gestión en una sola herramienta. Con Eventia, buscamos ofrecer una solución práctica y sencilla para la organización personal, facilitando la planificación y el control de actividades diarias.

Una de las características más destacadas de Eventia es su accesibilidad multilingüe. La aplicación estará disponible en varios idiomas, incluyendo inglés y español de manera predefinida, lo que ampliará su alcance y permitirá que más usuarios se beneficien de ella. Además, hemos diseñado la aplicación con un enfoque en la facilidad de uso. Queremos que personas de todas las edades, desde jóvenes hasta adultos mayores, puedan utilizar Eventia sin complicaciones, independientemente de su experiencia tecnológica.

En conclusión, con Eventia proporcionaremos a los usuarios una solución eficaz para la organización de su día a día. Nuestro objetivo es ayudarles a mejorar su productividad y gestión del tiempo, simplificando su vida día a día.

# 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

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: 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: 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.