

¡Hola!

Esta plantilla la creé para ayudarte a ti y los lectores que recién inician en el mundo del Desarrollo Android de la mano de [Hermosa Programación](#).

En ella no encontrarás terminología estricta de [Ingeniería de Software](#).

(Un estudio que agrega más nivel de detalle a la creación de software)

Pero si un plan ordenado para ejecutar paso a paso tareas que según yo, te vendrían bien si no tienes experiencia planificando proyectos.

Espero te sea de utilidad :)

[Aquí va el nombre de tu aplicación]

Escribir Una Descripción

Aquí va el objetivo general de tu aplicación, informando sobre sus características funcionales.

¿De qué forma impactará la vida del usuario?

Ej. Facilitará su comunicación verbal

¿Qué resultados obtendrá?

Ej. Conseguirá una reducción de tiempos de operación en un 20%

Investigar Quienes Serán Los Usuarios

Qué tipo de personas serán las que usen tu app. ¿Doctores, pacientes, vendedores, administradores de ventas?

Investiga que procesos cubrirás para esa persona.

Flujos de usuario

Una buena idea es detallar el flujo desde el inicio de la interacción hasta el final.

Por ejemplo...

Si tu usuario es un operario que toma medidas de una máquina, puedes detallar el proceso por pasos así:

1. El operario se desplaza al primer piso de las instalaciones
2. Observa el estado de su cable de conexión y anota su estado
3. Toma la temperatura de la parte A
4. Toma la temperatura de la parte B
5. Si lo considera, agrega evidencia fotográfica y agrega una descripción del suceso
6. Identifica si la temperatura supera un rango límite
 - a. Si es así, se notifica al supervisor
 - b. De lo contrario, se guardan las medidas para estadísticas mensuales

Con eso comprendes más de cerca que puntos de sus actividades estarás cubriendo.

Además, te ayudará a describir el siguiente punto.

Explorar Necesidades Del Usuario

Aquí van todas las características que tendrá la app. Es decir, sus funciones como sistema.

Algunos ejemplos:

- Guardar pedidos
- Grabar voces del usuario
- Tomar foto del incidente
- Mostrar lista de preguntas
- Filtrar proyectos por fecha
- Buscar usuarios por nombre
- Graficar estadísticas de polución
- ...

Los datos que se guardarán

En este paso es importante que identifiques las entidades del negocio a las cuales se le deben dar persistencia.

A lo mejor tu área del problema sea un inventario de productos químicos en tu tienda.

Y deseas que toda su información y la de los proveedores que te los venden, sea relacionada para guardarla en conjunto.

¿Dónde?

Bien, eso lo decidirás más adelante en la sección de **Fuentes de datos y estrategias de sincronización**.

Sin embargo aquí identificarás una lista de esas entidades, atributos y relaciones entre ellas.

Puedes usar [modelos de dominio](#), [diagramas entidad-relación](#), [diccionarios de datos](#), etc.

Nota: Hace un tiempo creé un ebook con técnicas para encontrar en tu problema las entidades a las cuales para las cuales debes guardar su información. Si quieres saber más ve al siguiente link.

[Click aquí para aprender más.](#)

Volviendo al ejemplo anteriormente mencionado...

Productos

Cada uno de los productos químicos que vendo. Por cada uno necesito tener registrado: código, unidades en inventario, imágenes, categoría, capacidad en litros, fecha de caducidad, el proveedor que me lo vende y el precio.

Proveedores

Son las personas jurídicas o naturales que me suministran los productos químicos. Por cada uno quiero guardar: número de identificación (NIT), nombre, razón social si aplica, representante legal, entre 1 y 3 datos de contactos directos si es empresa, dirección, teléfono, ubicación y productos que me suministra.

El formato anterior es la forma más básica posible de captura de datos para un negocio.

¡Pero funciona si aún no tienes conocimientos avanzados!

Crear Bocetos De La App

¿Cuál es el dispositivo físico objetivo?

Teléfonos, tablets, wearables, televisores o autos.

Dependiendo de tu caso, así mismo sabrás como crear sketches basados en patrones visuales acertados.

Incluye bosquejos sobre cómo debería verse tu app para satisfacer las necesidades del usuario.

Sumado a eso, escribe la cantidad de puntos de interacción para el usuario (que gesto realizará en cada control de la interfaz y que acción disparará)

¿Qué nivel de detalle debes usar?

Podría que solo necesites validar con tu grupo de trabajo y crea un dibujo en papel o un [wireframe](#) (baja fidelidad).

O que vayas a mostrar directamente el prototipo al cliente, usuario o inversionista de forma realista (alta fidelidad).

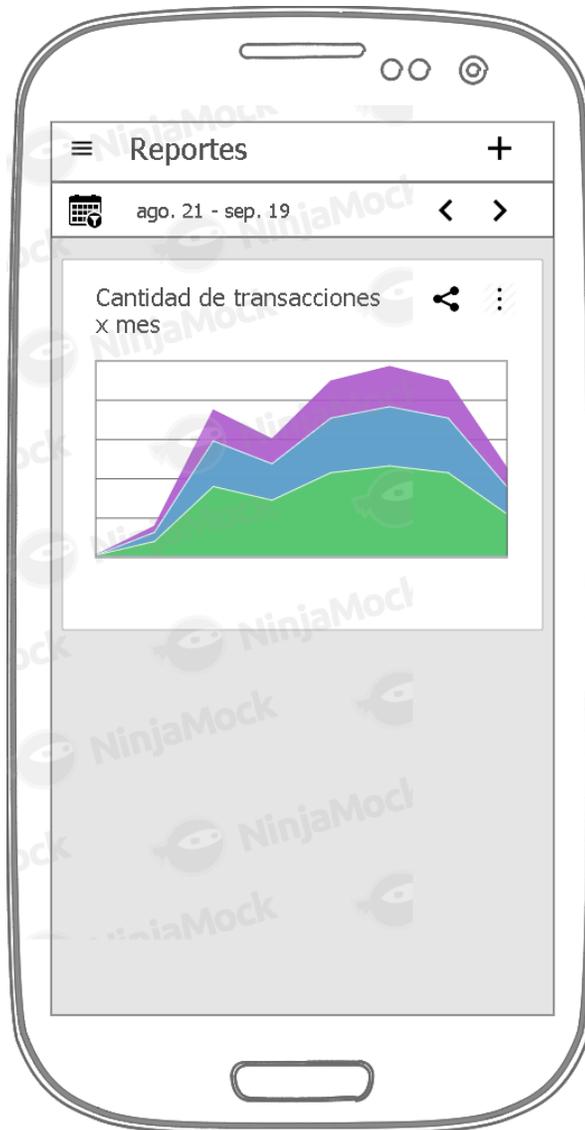
Incluso agregar interacciones visibles entre pantallas y componentes (Microinteracciones).

¿Qué herramientas emplear?

- [Fluid](#)
- [Indigo Studio](#)
- [Proto.IO](#)

- [NinjaMock](#)
- [Photoshop](#)
- [Balsamiq](#)
- [Mockplus](#)
- [Axure](#)
- [Invision](#)
- ...

Mira este ejemplo de wireframe...



Lo creé con NinjaMock. Muestra la representación de cómo obtener un gráfico con los reportes de transacciones de un negocio.

Sus **puntos de interacción** son:

- Icono del [Navigation Drawer](#): Abre menú horizontal
- [Action button](#) de signo más: Crea un nuevo reporte personalizado

- El texto del rango de fechas: Abre un menú para seleccionar el rango de fechas
- Los botones con flechas izquierda y derecha: Avanza o retrocede entre rangos de fechas consecutivos
- El botón de compartir: Envía una imagen PNG a una app compatible
- El botón de despliegue (tres puntos): Permite editar, duplicar y suprimir el gráfico.

Seleccionar Fuentes De Datos Y Estrategias De Sincronización

Llegado a este punto debes deducir que fuentes de datos usarás.

En otras palabras, de donde se leerán y almacenarán los datos:

- Bases de datos local propia
- Bases de datos nativas (*Contacts, Calendar, Documents, User Dictionary, etc.*)
- Bases de datos de un servidor propio
- APIs de terceros ([Google Maps](#), [Google Sheets](#), [Cloud Vision](#), etc.)
- La tarjeta SD del dispositivo
- Sistema de preferencias de Android
- La memoria del dispositivo
- ...

Ahora establece las reglas que seguirás para mantener las fuentes de datos sincronizadas.

Algunas preguntas a responder para darte inspiración son:

- ¿Qué operaciones de datos (lectura, guardado, actualización y eliminación) actúan sobre cada entidad detectada en el problema?
- ¿El usuario podrá sincronizar los datos manualmente?
 - ¿cuál es el tiempo necesario entre una sincronización y otra?
- ¿Cómo evitar que un registro que está pendiente por sincronizarse sea puesto de nuevo en cola?
- ¿Si el dispositivo no tiene conexión de red, de qué forma se controlará la sincronización?
- ¿Cuál es el pronóstico del volumen de datos que será sincronizado (ej. 1000 registros/día aprox.)?
- ¿Qué tan necesario crees que la sincronización debe ser automática y llevada a cabo a una hora determinada del día?
- ¿Cuál será el límite de registros que tendrás en tu base de datos local para evitar sobrecarga?
- ¿Si hay cambios en el servidor, se debe enviar una notificación a la app Android para sincronizar?

Elegir Recursos Y Herramientas

Investiga e incluye aquellas librerías, servicios, tecnologías, SDKs, APIs, etc. que debes considerar añadir a tu proyecto para solucionar problemas específicos.

Ejemplo:

- Bases de datos remotas: MySQL 5.7.8
- Lado servidor: PHP
- Formato de intercambio: JSON
- Formatos imágenes: PNG
- Cloud Computing: Google Cloud
- Gráficas Android: MPAndroidChart
- ...

¿De qué depende la elección?

De tus necesidades, contexto, presupuesto, talento humano, tiempo y varios factores más.

¿Desconoces que factores son vitales?

Investiga el funcionamiento del recurso y crea cuadros comparativos con sus propiedades.

Confrontando los pros y contras de cada opción, te acercará más a un veredicto acertado.

Por ejemplo, mira como [ProfitBricks](#) realiza una comparación de su servicio con [AWS de Amazon](#):

	Cores	RAM	Temporary SSD-Storage	Block Storage	Amazon Price EUR(hour)	Amazon Price EUR(30 days)	ProfitBricks Price EUR(hour)	ProfitBricks Price EUR(30 days)	Savings With ProfitBricks
M3 Medium Instance	1	3.75	4	250	0.078	56.40	0.0510	36.71	35%
M3 Large Instance	2	7.5	10	500	0.157	112.79	0.1021	73.50	35%
M3 Extra Large Instance	4	15	10	1.000	0.313	225.58	0.2036	146.60	35%

Comparan varios atributos de las instancias de máquinas virtuales como núcleos, RAM, almacenamiento SSD, y block storage.

En la última con columna se determina que **ProfitBricks** ofrece mejores precios al agrupar una serie de características.

Intentando ayudar a las personas que aún no se deciden en su compra.

Versión SDK

Además determinar cuál será la versión mínima y la versión objetivo del SDK de Android.

Ejemplo:

Una app que soporta versiones igual o mayores a 17 (Android Jelly Bean 4.2) y su objetivo de despliegue es 21 (Android Lollipop 5.0).

Idiomas Soportados

¿En cuántos idiomas tendrás que traducir los mensajes de la interfaz y servicios asociados?

Tenlo en mente. Este es otro costo para la app.

Desarrollar Una Versión Cascarón

Le llamo cascaron a una app que usa datos y señales de entrada falsos.

En esta fase solo te enfocarás en desarrollar las interacciones de UI sin tener en cuenta dependencias externas reales.

Para ello debes listar tareas de programación asociadas a las funcionalidades que desarrollarás.

Por ejemplo:

1. Crear y configurar proyecto en Android Studio
 - a. Añadir librerías
 - b. Definir estructura de paquetes Java
 - c. Otra
 - d. ...
2. Iniciar un punto de versionamiento con Github
 - a. Crear cuenta en GitHub
 - b. Crear repositorio
 - c. Otra
 - d. ...
3. Implementar login
 - a. Crear actividad actividad de login
 - b. Crear fragmento de login
 - c. Otra
 - d. ...
4. Otra
5. ...

La extensión de la tarea, el orden y la clasificación (UI, Lógica de negocios, datos, servicios, etc.) depende de tu metodología de resolución de problemas.

Conocimientos Prerrequisitos

Ahora, si aún no sabes cómo usar elementos de programación en Android, entonces aparta tiempo y planificación para aprenderlo.

En otras palabras, **planifica tareas para obtener el conocimiento.**

Digamos que no sabes todavía manejar actividades. Una buena forma de expresarlo en tu lista de tareas sería:

1. Implementar Login
 - a. Aprender a crear instancias de la clase `Activity` (`AppCompatActivity` para soporte)
 - b. Entender el diseño a través de [Layouts](#)
 - c. Conocer el funcionamiento básico de la Action Bar.
 - d. ...

Ten en cuenta estos elementos a la hora de calcular tu tiempo de desarrollo.

Implementar fuentes de datos

Finalmente con una app puesta en marcha, creas las respectivas implementaciones para operar con datos reales.

Tareas como crear tu [base de datos SQLite](#), [montar tu servicio web](#), añadir controladores para la [lectura de ubicaciones](#), etc. son las que hacen parte de este punto final.

¿Cómo Te Fue?

¿Cómo viste esta plantilla?

¿Crees que te traerá beneficios para establecer un norte en tu proyecto Android?

Espero que así sea.

Ahora el paso a seguir es que le hagas una copia con el nombre

Documento De Planificación Para [Nombre App].docx

Luego edita con tus propias definiciones sin complicarte.

Es casi seguro que tu plan cambiará constantemente. Así que no te opongas a ello, solo adáptate.

Finalmente...

Ve **AQUÍ** y hazme saber a mí y los demás lectores, si esta plantilla te resultó útil y si valió la pena descargarla.

Si tienes recomendaciones como agregar más pasos, corregir errores, modificar términos o demás, envíame un correo a contacto@hermosaprogramacion.com con tu apreciación.

Además, si eres nuevo en [Hermosa Programación](#), déjame saber un poco app estás desarrollando en este momento.

Como supondrás, me llegan toneladas de mensajes a mi correo cada día, y aunque intento responderlos todos, es casi imposible.

Así que no te sientas mal si no te respondo, sin importar qué, lo leeré para tenerlo en cuenta en mis nuevos contenidos.

Saludos,
James