

Introducción al Desarrollo de Aplicaciones

- Ing. Skrauba Axel

¿Qué es el desarrollo de aplicaciones?

Proceso de crear programas de software diseñados para cumplir tareas específicas y resolver necesidades concretas de los usuarios.

Estas aplicaciones pueden ejecutarse en diferentes dispositivos y plataformas, como celulares, tablets, computadoras o navegadores web.

¿Qué es el desarrollo de aplicaciones?

Proceso de crear programas de software diseñados para cumplir tareas específicas y resolver necesidades concretas de los usuarios.

Diseño de Software

Proceso en el cual creamos componentes y los relacionamos de forma óptima para cumplir un objetivo.

¿Qué es el desarrollo de aplicaciones?

Proceso de crear programas de software diseñados para cumplir tareas específicas y resolver necesidades concretas de los usuarios.

Requerimientos

Los hay de varios tipos, y no todos están vinculados directamente a resolverle la necesidad al usuario.

- Funcionales
- No funcionales

Tipos de requerimientos

Funcionales

¿Qué debe hacer la aplicación?

¿Qué debería de poder hacer el usuario?

Características que describen el funcionamiento.

- Ej.: Poder dar *like* a un post.

No Funcionales

Características específicas, intrínsecas. Límites de la solución.

Ej.: rendimiento, seguridad, calidad.
Utilizar tantos MB como máximo.

Ejemplos de tipos de requerimientos

- El usuario debe poder hacer *login*.
- La espera para los datos del servidor no debe superar 10 segundos.
- Se debe guardar la información por 5 días.
- El usuario puede cambiar su foto de perfil.
- El usuario debe navegar sin conexión.

■ Funcional

■ No Funcional

■ No Funcional

■ Funcional

■ No Funcional

Tipos de aplicaciones

De Escritorio

Estas aplicaciones se ejecutan en computadoras de escritorio o notebooks y se instalan directamente en el sistema operativo del dispositivo. Suelen ofrecer una amplia gama de funciones y pueden ser utilizadas sin conexión a Internet.

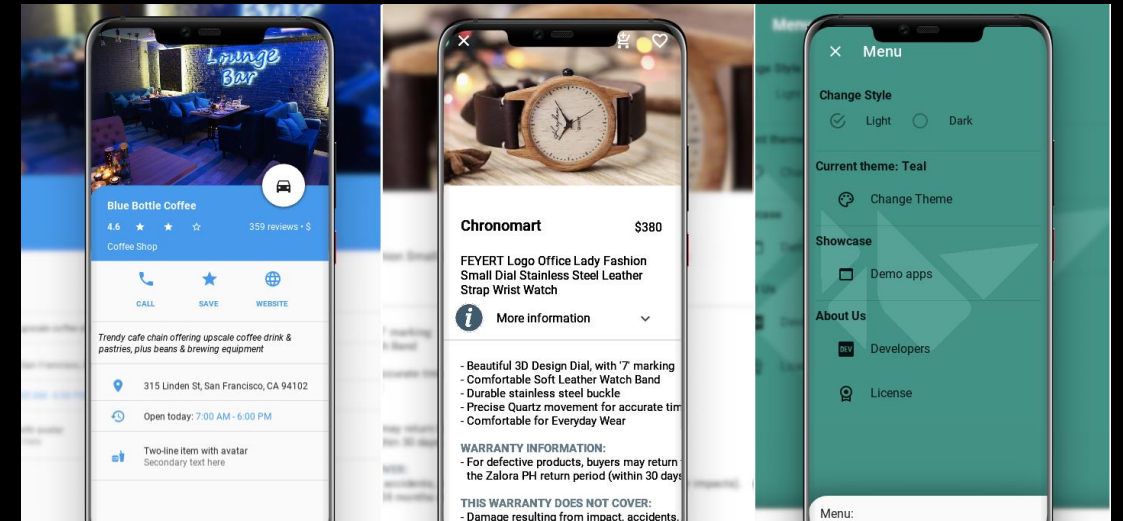
WEB

Son accesibles a través de navegadores web y no requieren instalación en el dispositivo del usuario. Estas aplicaciones suelen ser multiplataforma y se actualizan de forma centralizada, lo que facilita el mantenimiento y la distribución.

Tipos de aplicaciones

Móviles

Son programas diseñados para funcionar en dispositivos móviles como celulares y tablets. Pueden ser aplicaciones nativas, desarrolladas específicamente para una plataforma particular, como Android o iOS, o aplicaciones web que se ejecutan a través de navegadores móviles.



El proceso de desarrollo de aplicaciones

Análisis y planificación

Se definen los **objetivos** y **requisitos** de la aplicación, se identifica el **público objetivo** y se establece una **estrategia** para su desarrollo.

Diseño de la aplicación

Se crea una **representación visual** de la interfaz de usuario y se define la arquitectura general de la aplicación, incluyendo la estructura de datos y la lógica de funcionamiento.

El proceso de desarrollo de aplicaciones

Desarrollo

Se **escribe** el **código** de la aplicación utilizando el lenguaje de programación y las herramientas seleccionadas. Aquí es donde se da vida a la aplicación y se implementa su funcionalidad.

Pruebas y depuración

Se realizan **pruebas exhaustivas** para **identificar y corregir errores** y fallos en la aplicación.

El proceso de desarrollo de aplicaciones

Lanzamiento y distribución

Una vez que la aplicación ha sido probada y optimizada, se lanza y **distribuye a los usuarios** finales a través de tiendas de aplicaciones o en la web.

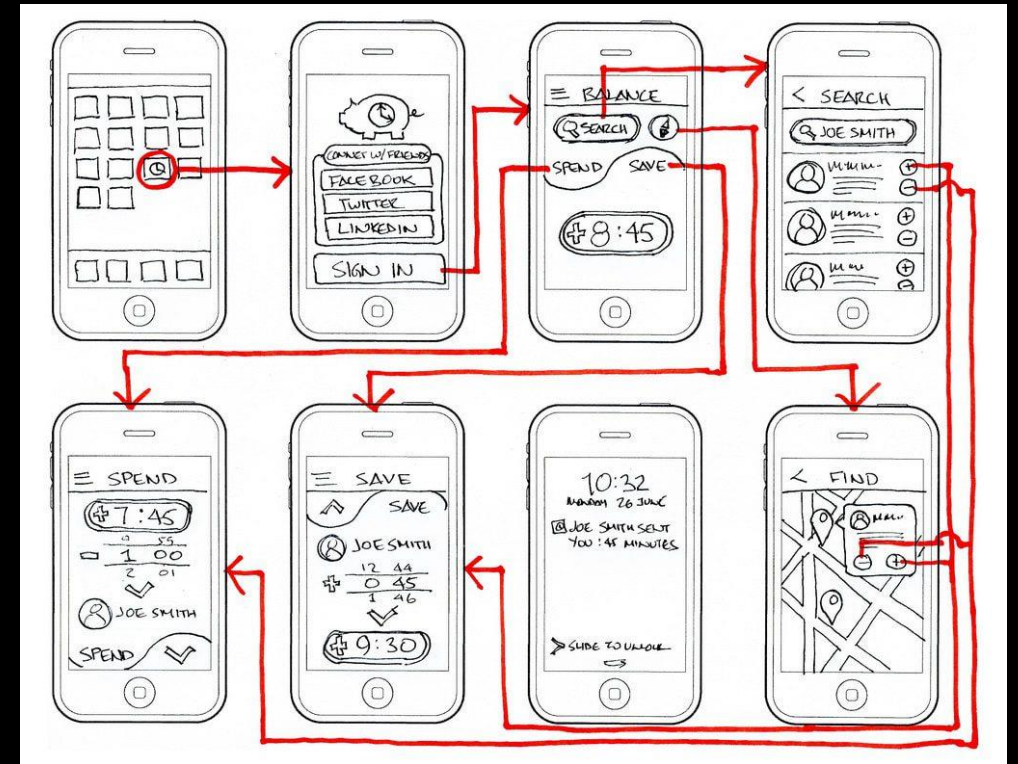
Mantenimiento y actualización

Después del lanzamiento, es esencial **mantener la aplicación**, realizar actualizaciones periódicas y responder a las necesidades cambiantes de los usuarios.

Elementos básicos de una aplicación

Interfaz de usuario (UI)

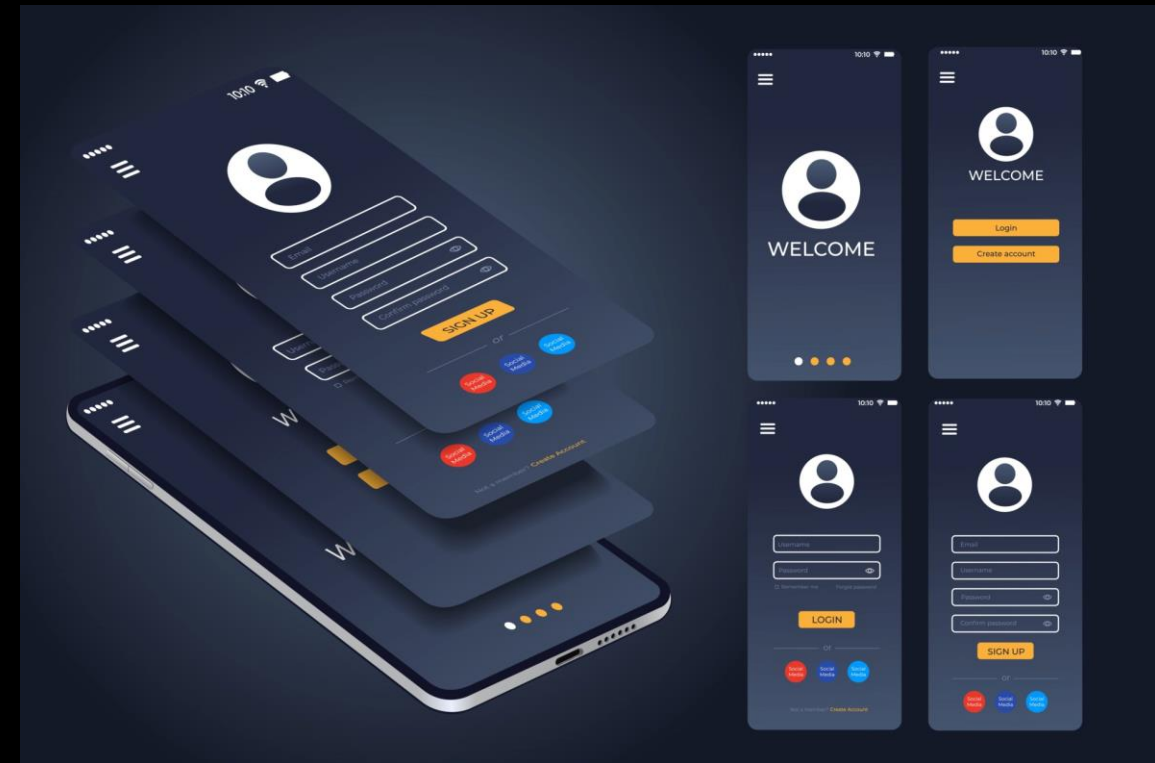
Es el medio por el cual los usuarios interactúan con la aplicación. Puede incluir elementos visuales como botones, campos de entrada, menús desplegables y otros componentes que permiten a los usuarios realizar acciones y obtener resultados. La UI debe ser intuitiva, atractiva y fácil de usar **para mejorar la experiencia del usuario.**



Elementos básicos de una aplicación

Interfaz de usuario (UI)

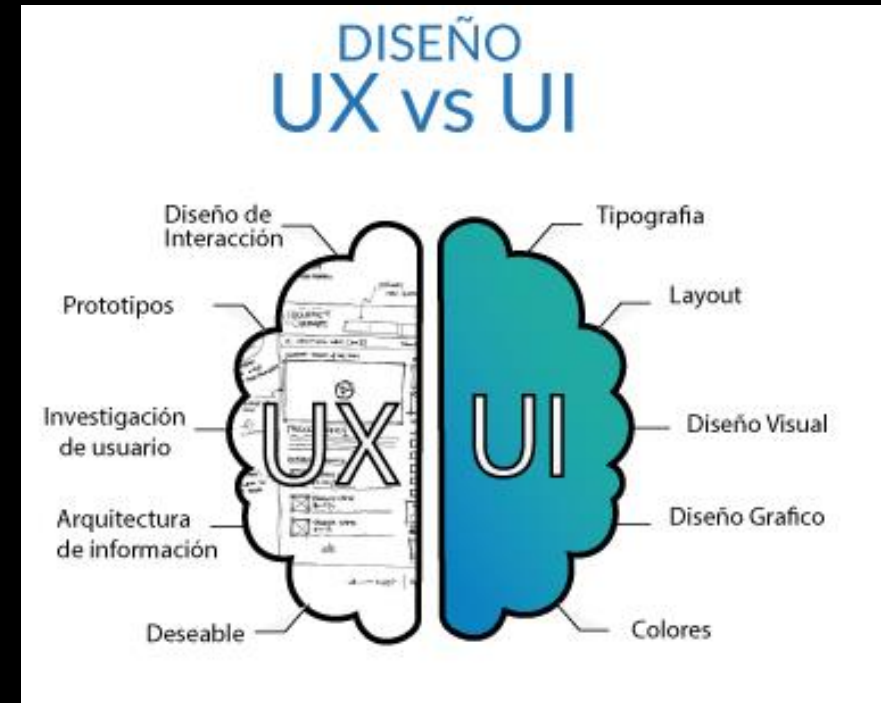
Es el medio por el cual los usuarios interactúan con la aplicación. Puede incluir elementos visuales como botones, campos de entrada, menús desplegables y otros componentes que permiten a los usuarios realizar acciones y obtener resultados. La UI debe ser intuitiva, atractiva y fácil de usar **para mejorar la experiencia del usuario.**



Elementos básicos de una aplicación

Experiencia de usuario (UX)

Se define en función de cuan fácil o difícil es para el usuario interactuar con el producto. Por lo general se apunta a un **producto usable, fácil e intuitivo**.



Elementos básicos de una aplicación

Experiencia de usuario (UX)

Se define en función de cuan fácil o difícil es para el usuario interactuar con el producto. Por lo general se apunta a un **producto usable, fácil e intuitivo**.



Elementos básicos de una aplicación

Lógica de programación

Es el conjunto de instrucciones y algoritmos que controlan el comportamiento de la aplicación. Aquí es donde se define cómo debe responder la aplicación ante las acciones del usuario y cómo debe procesar los datos para producir resultados específicos.

```
class Container(BoxLayout):
    """A container is essentially a class that loads its root from a known
    .kv file.

    The name of the .kv file is taken from the Container's class.
    We can't just use kv rules because the class may be edited
    in the interface and reloaded by the user.
    See :meth: change_kv where this happens.
    """

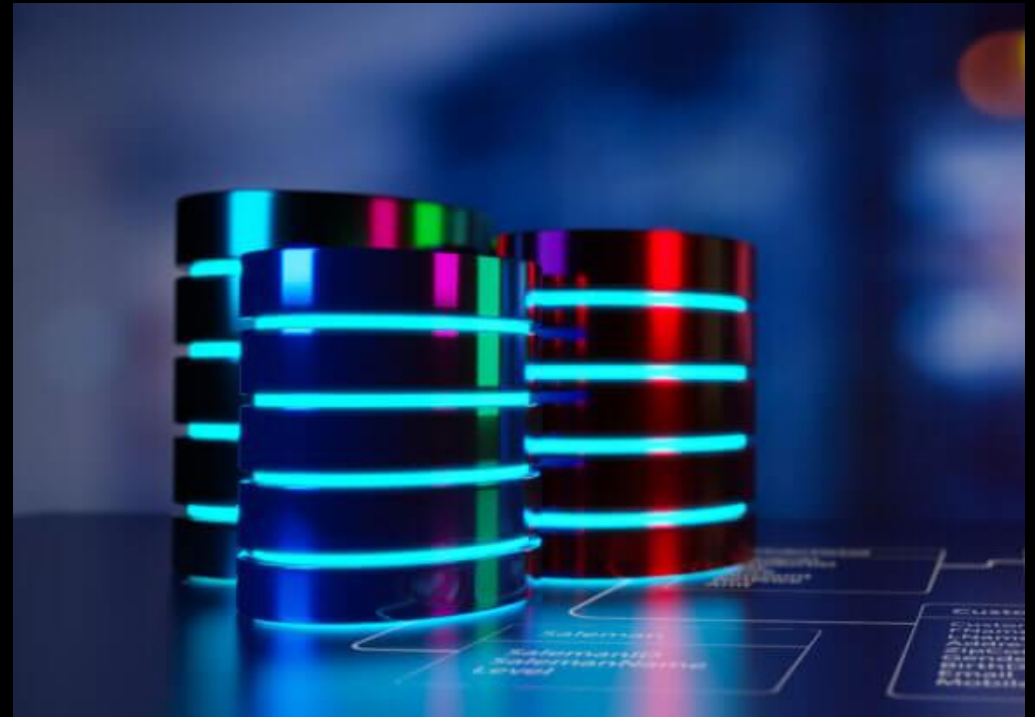
    def __init__(self, **kwargs):
        super(Container, self).__init__(**kwargs)
        self.previous_text = open(self.kv_file).read()
        parser = Parser(content=self.previous_text)
        widget = Factory.get(parser.root.name)()
        Builder._apply_rule(widget, parser.root, parser.root)
        self.add_widget(widget)

    @property
    def kv_file(self):
        """Get the name of the kv file, a lowercase version of the class
        name.
        """
        return os.path.join(CONTAINER_KVS, self.__class__.__name__ + '.kv')
```


Elementos básicos de una aplicación

Almacenamiento de datos

Las aplicaciones generalmente requieren guardar información importante, como perfiles de usuario, configuraciones, registros, etc. Esto se puede lograr utilizando bases de datos, archivos locales o incluso servicios en la nube, dependiendo de la naturaleza de la aplicación.



Elementos básicos de una aplicación

Seguridad y privacidad

Es fundamental proteger la seguridad y privacidad de los usuarios y sus datos. Las aplicaciones deben implementar medidas de seguridad para proteger información sensible y evitar vulnerabilidades, como la inyección de código malicioso o el acceso no autorizado a los datos.



Elementos básicos de una aplicación

Funcionalidades específicas

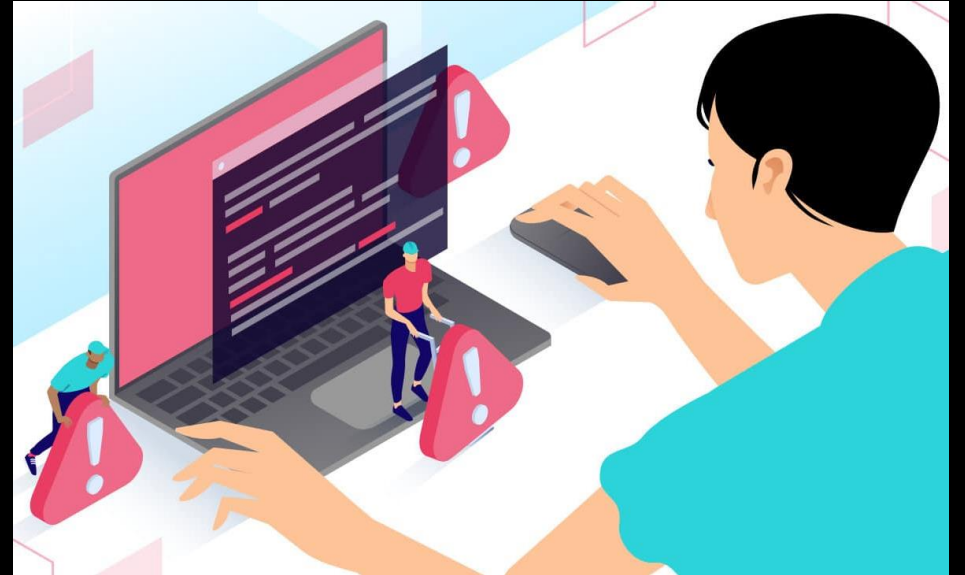
Son las que resuelven la necesidad particulares del usuario. Puede ser desde una simple calculadora hasta una aplicación de redes sociales. La implementación de estas funcionalidades es el núcleo de la aplicación y suele estar respaldada por la lógica de programación y el almacenamiento de datos.



Elementos básicos de una aplicación

Manejo de errores y excepciones

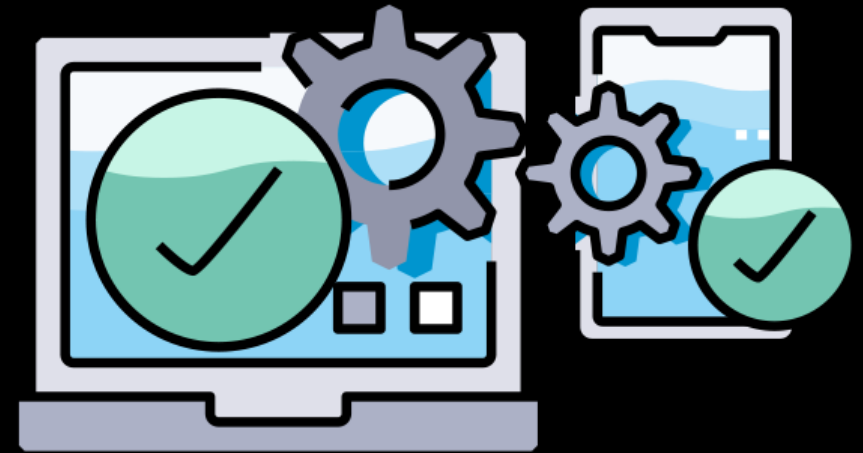
Las aplicaciones deben estar preparadas para manejar errores y excepciones que puedan surgir durante su ejecución. Esto incluye situaciones como conexiones a Internet interrumpidas, datos incorrectos ingresados por el usuario o cualquier otro escenario inesperado. Un buen manejo de errores mejora la robustez y confiabilidad de la aplicación.



Elementos básicos de una aplicación

Actualizaciones y mantenimiento

Una vez lanzada, una aplicación debe recibir actualizaciones periódicas para corregir errores, agregar nuevas características o mejorar su rendimiento. El mantenimiento adecuado asegura que la aplicación siga funcionando correctamente en un entorno en constante cambio.



Lenguaje de programación y entorno de desarrollo

El desarrollo de aplicaciones involucra el uso de lenguajes de programación y entornos de desarrollo, que son herramientas esenciales para escribir código y crear software funcional.

Existen numerosos lenguajes de programación y entornos, cada uno con sus características y ventajas.

Lenguaje de programación y entorno de desarrollo

Lenguaje

Python: Es uno de los lenguajes más populares para principiantes debido a su sintaxis sencilla y legible. Es versátil y se utiliza ampliamente en el desarrollo web, análisis de datos, inteligencia artificial y automatización de tareas.



Entorno

Visual Studio Code: Es un IDE gratuito y altamente recomendado para principiantes. Tiene soporte para múltiples lenguajes y ofrece extensiones que facilitan el desarrollo de aplicaciones.



Framework

¿Qué es?

Un **entorno de trabajo** o **marco de trabajo** es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

¿Ventajas?

Utilizar un *framework* permite **agilizar los procesos** de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas **buenas prácticas** y la **consistencia** del código.

Kivy

The Open Source Python App Development Framework.

Build and distribute
beautiful Python cross-
platform GUI apps with
ease.

¿Ventajas?

Kivy ha sido creado para ser fácil de usar, multiplataforma y rápido.

Con un solo código base, podrá implementar aplicaciones en Windows, Linux, macOS, iOS y Android.

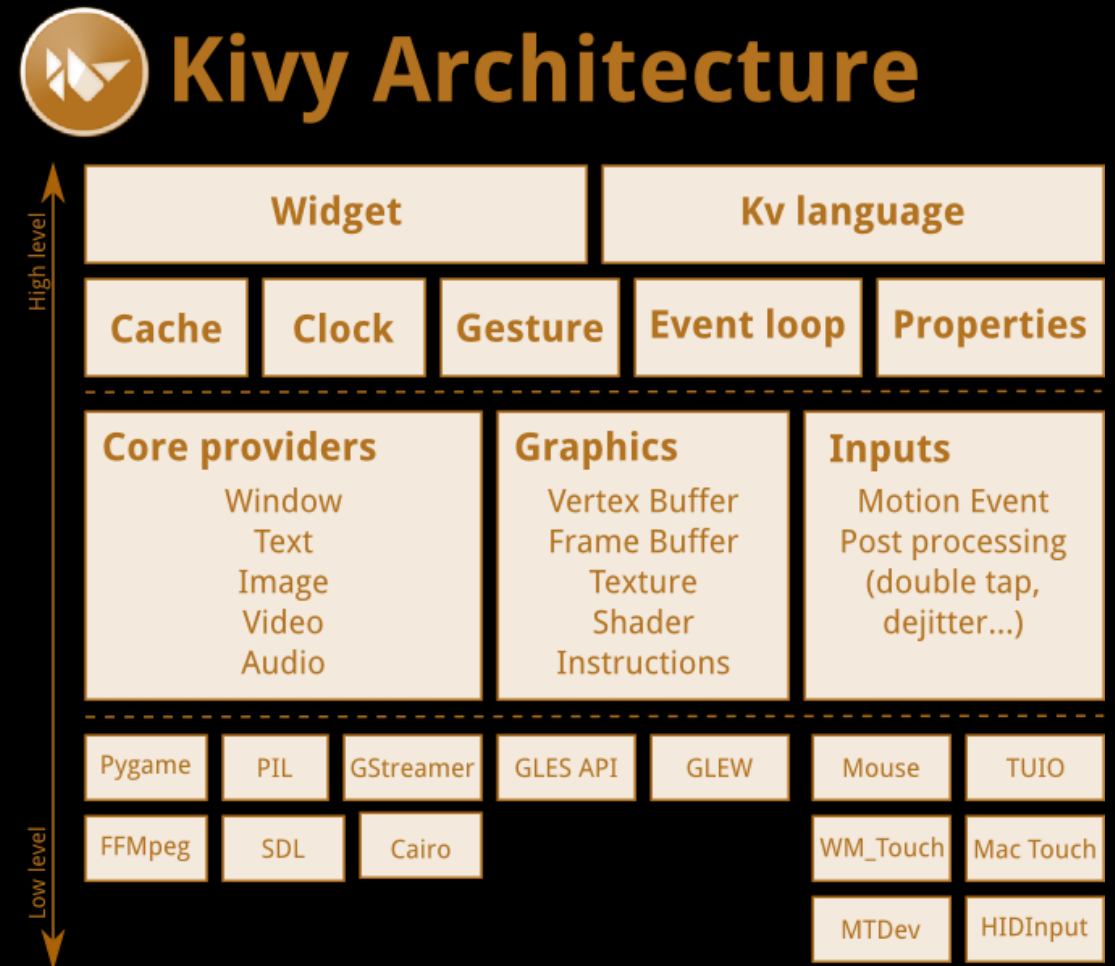


<https://kivy.org/>

Kivy: arquitectura

Diseño de *Kivy* desde el punto de vista de la ingeniería de software.

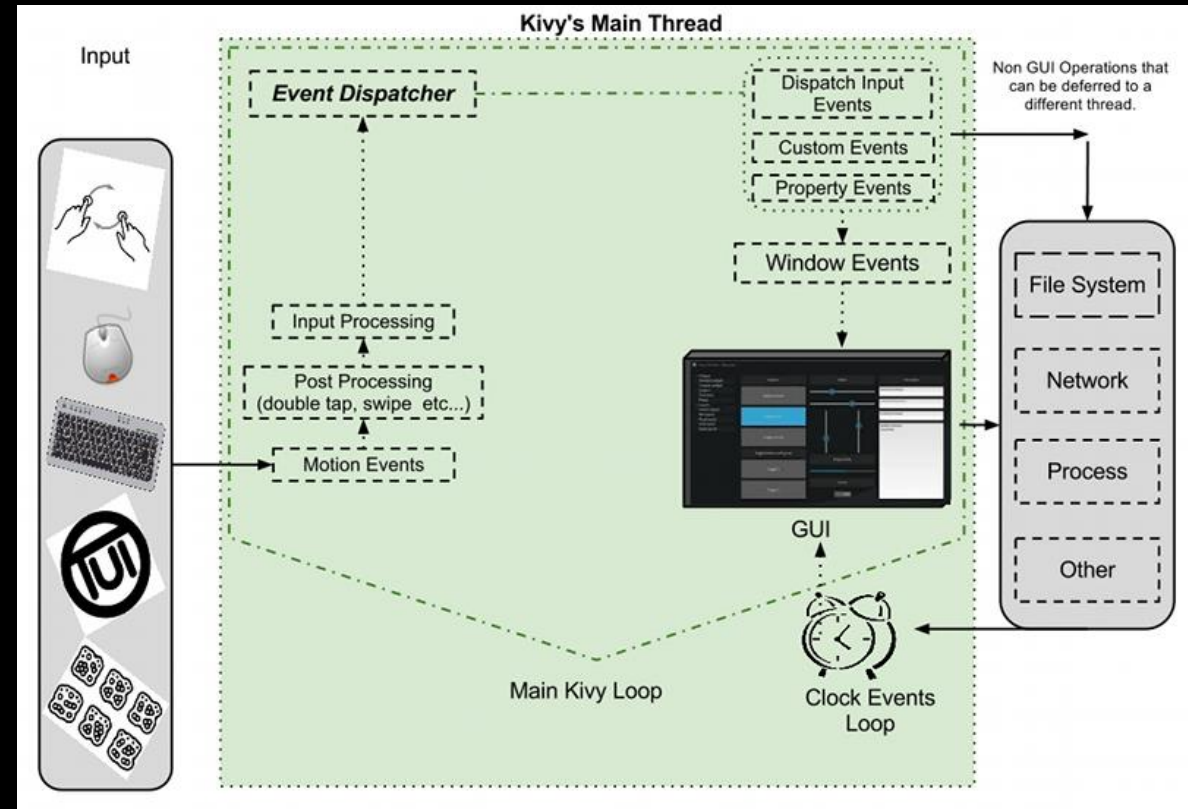
<https://kivy.org/doc/stable/guide/architecture.html>



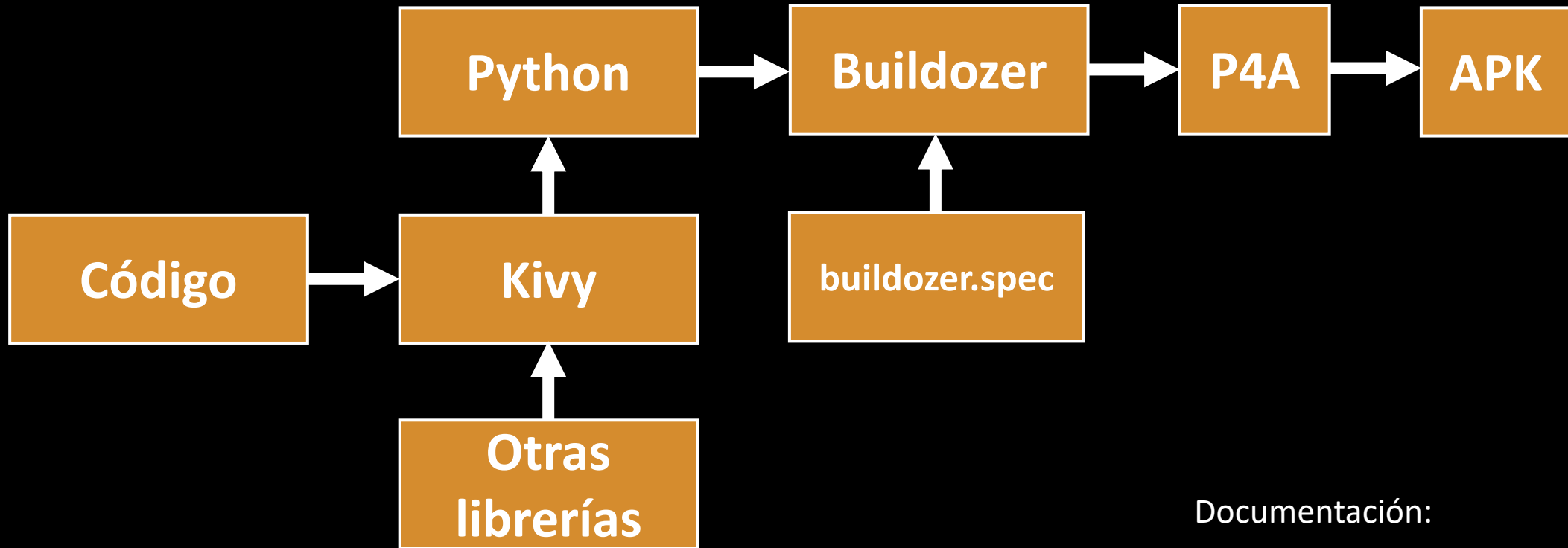
Kivy: gestión de eventos

Los eventos son una parte importante de la programación de Kivy. Eso puede no ser sorprendente para aquellos con experiencia en desarrollo de GUI, pero es un concepto importante para los recién llegados.

<https://kivy.org/doc/stable/guide/events.html>



Kivy: ¿Cómo obtengo un *.apk*?



Documentación:

- <https://kivy.org/doc/stable/>
- <https://github.com/kivy/buildozer>

Preguntas...

- Dudas
- Sugerencias
- Cuestiones

Sino... a conocer
un poco el futuro
espacio de trabajo.