

Herramientas y Tecnologías

Kairos

NexTech

Centurión Valeria, Escalante Guillermo, Maldonado Agustina, Mendez Florencia, Ulloa Gonzalo.

**

**



**Tabla de contenido**

[Introducción 4](#_meg1zv33sqdo)

[Herramientas y tecnologías 4](#_r3c94xrxn9io)

[*Lenguajes de programación 4*](#_1x3fypkwr9f2)

[*Framework 4*](#_jg6kdovnp1av)

[*Bases de datos 4*](#_ahmla9ug5eqk)

[*Herramientas de prueba 4*](#_7jj9801sxdpb)

[*Herramientas de desarrollo. 4*](#_u8dwi1xqv8z)

[*Herramientas de documentación 4*](#_vf9v28czmzhz)

[*Otras 5*](#_do7lbd6gfuno)

Herramientas y tecnologías

# Introducción

El presente documento tiene como propósito definir y detallar las herramientas que serán

utilizadas para el desarrollo del software “Kairos”.

# Herramientas y tecnologías

## Lenguajes de programación

Para el desarrollo del Backend de la aplicación se empleará el lenguaje de programación Java.

Para el desarrollo del Frontend de la aplicación se empleará el lenguaje de programación

TypeScript.

## Framework

Se utilizarán los siguientes framework para el desarrollo de la aplicación:

* SpringBoot
* Angular

## Bases de datos

Se utilizará MySql para la creación de bases de datos y phpMyAdmin para la gestión de las

mismas.

## Herramientas de prueba

Para las pruebas unitarias se utilizaran:

* JUnit
* Mockito
* Postman

## Herramientas de desarrollo Se utilizará el software VisualStudio Code como entorno de desarrollo. Para la gestión de versiones, se utilizara Git, GitHub y Rclone.

## Herramientas de documentación

Se utilizará Javadoc y TypeDoc para llevar a cabo la documentación del proyecto.

## Otras

Otras herramientas que se utilizarán durante el proyecto:

* Herramientas de Google (docs, hojas de cálculo, etc)
* Canva
* Discord
* WhatsApp
* TogglTrack
* [Draw.io](http://draw.io)
* Docker

## Conclusión

La selección de herramientas y tecnologías es un paso crítico para garantizar el éxito del

proyecto, ya que influye directamente en la eficiencia del desarrollo, la calidad del

producto final y la facilidad de mantenimiento a largo plazo.

Tras el análisis de las distintas opciones disponibles, se han identificado aquellas que mejor se adaptan a los requerimientos funcionales y no funcionales del proyecto, considerando factores como compatibilidad, escalabilidad, soporte, curva de aprendizaje y costo.