Logotipo

Descripción generada automáticamente

**Universidad Tecnológica De Nezahualcóyotl.**

**ING. EN DESARROLLO Y GESTIÓN DE SOFTWARE**

**integrantes:**

GUTIERREZ URIOSTEGUI CINTHIA LIZBETH

LAZCANO MARTINEZ ALAIN ENRICO

LUNA APOLINAR LUIS ALBERTO

MORA MONTES JAVIER SALATIEL

**Grupo:**

ITIC1001-V

**Proyecto:**

MusicBar

Índice

[Plan del proceso de desarrollo 3](#_Toc99539589)

[Justificación de la arquitectura 3](#_Toc99539590)

[Diagrama de arquitectura 4](#_Toc99539591)

[Patrones de diseño 5](#_Toc99539592)

[Justificación de Framework 5](#_Toc99539593)

[Cronograma 7](#_Toc99539594)

[Product Backlog 7](#_Toc99539595)

[**Modelo Canvas** 10](#_Toc99539596)

[Diagramas de actividad Musicbar 11](#_Toc99539597)

[**Documentación de cambio de proceso de desarrollo a DevOps** 14](#_Toc99539598)

[Introducción 14](#_Toc99539599)

[Planeación 14](#_Toc99539600)

[Alcance 16](#_Toc99539601)

[Arquitectura 17](#_Toc99539602)

[Code 18](#_Toc99539603)

[Desarrollo 18](#_Toc99539604)

[Build 19](#_Toc99539605)

[Test 20](#_Toc99539606)

[Release 21](#_Toc99539607)

[Deploy 21](#_Toc99539608)

[Operate 22](#_Toc99539609)

[Monitor 23](#_Toc99539610)

[En la fase de monitoreo 23](#_Toc99539611)

# Plan del proceso de desarrollo

Se migrará una parte de la aplicación web previamente desarrollada a plataforma móvil, por lo cual, se hará la investigación pertinente para poder lograr el objetivo planteado desde un inicio, se adaptará la arquitectura que mejor se adapte al desarrollo y a los procesos que conformaran a la aplicación móvil.

También se analizará un framework de desarrollo para plataformas móviles para hacer la migración. Analizando los puntos fuertes o que mejor se adapten a la necesidad del desarrollo del proyecto.

Es importante identificar los módulos que serán adaptados y documentar todos los movimientos que se vayan haciendo para futuras actualizaciones o modificaciones a la aplicación.

# Justificación de la arquitectura

Las arquitecturas monolíticas son otro tipo de arquitectura asociado con los sistemas heredados; son pilas de aplicaciones únicas que contienen todas las funciones dentro de cada aplicación. Tienen conexión directa, tanto en la interacción entre los servicios como en la manera en que se desarrollan y distribuyen.

Ventajas:

* Los programas son fáciles de desarrollar.
* El despliegue y la ejecución del software son muy sencillos.
* El costo de desarrollo es bajo en comparación con otras arquitecturas.

# Microservicios - Comparativa CloudsDiagrama de arquitectura

# Patrones de diseño

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización.

Diagrama

Descripción generada automáticamente

# Justificación de Framework

Kotlin es un proyecto gratuito y de código abierto registrado bajo la licencia de Apache 2.0. El código del proyecto se desarrolla abiertamente en GitHub y está a cargo, principalmente, del equipo empleado en JetBrains, con contribuciones de Google y otros.

Elegimos Kotlin, ya que cuenta con ecosistema de desarrollo abierto a medida que evoluciona se agranda la plataforma de Android, aparte cuenta con una gran comunidad, documentación y videos para poder comprender mejor el Framework, a su vez es compatible con las herramientas de desarrollo previamente utilizadas.

**Ventajas**

* Menos código combinado con mayor legibilidad. Pase menos tiempo escribiendo su código y trabajando para comprender el código de los demás.
* Compatibilidad con Kotlin en Android Jetpack y otras bibliotecas . [Las extensiones KTX](https://developer.android.com/kotlin/ktx) agregan características del lenguaje Kotlin, como rutinas, funciones de extensión, lambdas y parámetros con nombre, a las bibliotecas de Android existentes.
* Código de seguridad. Menos código y una mejor legibilidad conducen a menos errores. El compilador de Kotlin detecta estos errores restantes, lo que hace que el código sea seguro.
* Fácil aprendizaje. Kotlin es muy fácil de aprender, especialmente para los desarrolladores de Java.
* Gran comunidad. Kotlin cuenta con un gran apoyo y muchas contribuciones de la comunidad, que está creciendo en todo el mundo. Según Google, más del 60% de las 1000 aplicaciones principales en Play Store usan Kotlin.



# Cronograma

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente



# Product Backlog

Tabla

Descripción generada automáticamente

|  |  |  |  |
| --- | --- | --- | --- |
| Framework | Definición | Ventajas | Desventajas |
| Cacharreando Kotlin – HackLab Almería | Kotlin es un nuevo lenguaje de programación funcional, desarrollado por el equipo de JetBrains, el cuál ha ido tomando peso, debido a que entró a formar parte oficialmente como lenguaje Android en 2017, proporcionando ventajas sobre Java 6. Esto se debe a que se compila a bytecode JVM. | \*Ofrece todas las características de un lenguaje de estilo orientado a objectos con un enfoque funcional  Funciones de orden superior, son aquellas que toman una función como argumento y tipo de retorno  \*La codificación en Kotlin es muy corta, concisa y directa. Mismo trabajo con menos líneas de código que en Java  \*Respaldado por JetBrains, Google y Spring Framework, definitivamente está aquí para quedarse, crecer y evolucionar | \*Coincidencia de patrones débil  \*Tamaño extra de tiempo de ejecución  \*Legibilidad inicial del código se vuelve un poco difícil de leer y entender al principio  \*Comunidad de soporte pequeña, Kotlin es más joven que Scala, tiene menos bibliotecas, publicaciones en blogs y tutoriales |
| Flutter - Build apps for any screen | Flutter es una mezcla entre un framework y un Software Development Kit (SDK) que está construido en C y C++ para programar en lenguaje Dart.  Flutter compila tanto a Android (NDK) como IOS (LLVM). Además, Flutter trae su propio motor de renderizado, llamado Skia, el cual se incluye en la propia aplicación compilada,  lo que le da una velocidad constante de 60 fps. | \*Supera en prestaciones a todos los frameworks de desarrollo multiplataforma existentes  \*Hot reload que facilita el desarrollo.  \*Renderizado de vistas muy rápido y constante (60fps).  \*Programas en un solo lenguaje y compilado para 2 plataformas diferentes, lo que permite ahorrar tiempo y recursos.  \*Acceso al hardware de los móviles tales como la cámara, geolocalización, red, almacenamiento o a través de SDK’s de terceros. | \*Aprender el lenguaje Dart desde cero.  \*Por ahora solo se puede desarrollar aplicaciones en 2D.  \*Framework muy joven con poca comunidad, por lo que deberás afrontar los problemas que te encuentres con menos ayuda que en otros frameworks.  \*Está enfocado solo a móvil. |
| Xamarin frente a React Native | Secretos desbloqueados | Xamarin es un software imprescindible para el desarrollo de aplicaciones móviles, puesto que permite a los desarrolladores compartir hasta un 90% del código entre plataformas distintas, utilizando un único lenguaje de programación, C#, y que cada vez está ganando más adeptos tanto por el rendimiento como por la posibilidad del acceso al API nativo. | \*Una pila de tecnología para codificar para todas las plataformas  \*Xamarin usa C # complementado con .Net framework para crear aplicaciones para cualquier plataforma móvil. Por lo tanto, puede reutilizar hasta el 96 por ciento del código fuente acelerando el ciclo de ingeniería.  \*Rendimiento cercano al nativo  \*Experiencias de usuarios nativos  \*Soporte completo de hardware  \*Xamarin admite enlaces con bibliotecas nativas  \*Tecnología de código abierto con un fuerte soporte corporativo  \*Mantenimiento simplificado  \*Ecosistema de desarrollo completo | \*Soporte ligeramente retrasado para las últimas actualizaciones de plataforma  \*Acceso limitado a bibliotecas de código abierto  \*El desarrollo nativo hace un uso extensivo de las tecnologías de código abierto.  \*Alto costo para uso profesional y empresarial |

# **Modelo Canvas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Aliados Clave**  Proveedores de servicios en la nube. | **Actividades Clave**  Estudio de mercado  Buscar socios  Mercadotecnia | **Propuesta de Valor**  El software permite que los usuarios finales puedan realizar reservaciones, ya que se despliega un mapa donde se muestra los lugares disponibles. Adicionalmente se pueden subir las promociones y los eventos del establecimiento. | | **Relación con el Cliente**  Se entregará el manual de usuario, adicionalmente se capacitará al personal para que pueda hacer uso de la plataforma.  Se mantendrá comunicación para brindar asesorías. | **Segmentos de Clientes**  Establecimientos que cuenten con la modalidad de reservación.   * Bares * Restaurantes * Establecimientos de Comida * Bar-restaurante   -Música en vivo. |
| **Recursos Clave**  Personal   * Equipo de TI * Mercadotécnicos   Tecnología   * Hosting * Licencias * Equipos de computo   Lugar de trabajo | **Canales:**   * Correo Electrónico * Página Web * WhatsApp * Teléfono |
| **Estructura de Costos**  Diseñadores, programadores, hosting, seguridad.  Gestión  Mercadotécnico  Infraestructura | | | **Estructura de Ingresos**  **Sistema de suscripción**  **Formas de pago:** Efectivo  Tarjeta crédito/ debito  Transferencias | | |

# Diagramas de actividad Musicbar

Diagrama de actividad Login

Diagrama

Descripción generada automáticamente

Diagrama de actividad Registro

Diagrama

Descripción generada automáticamente

Diagrama de actividad Reservación

Diagrama, Esquemático

Descripción generada automáticamente

# **Documentación de cambio de proceso de desarrollo a DevOps**

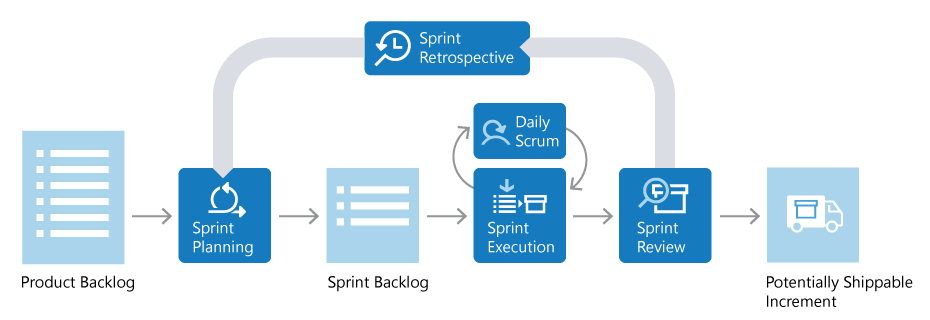
# Introducción

En el presente documento se describirán los métodos implementados para el desarrollo del proyecto Music Bar bajo el enfoque de DevOps.

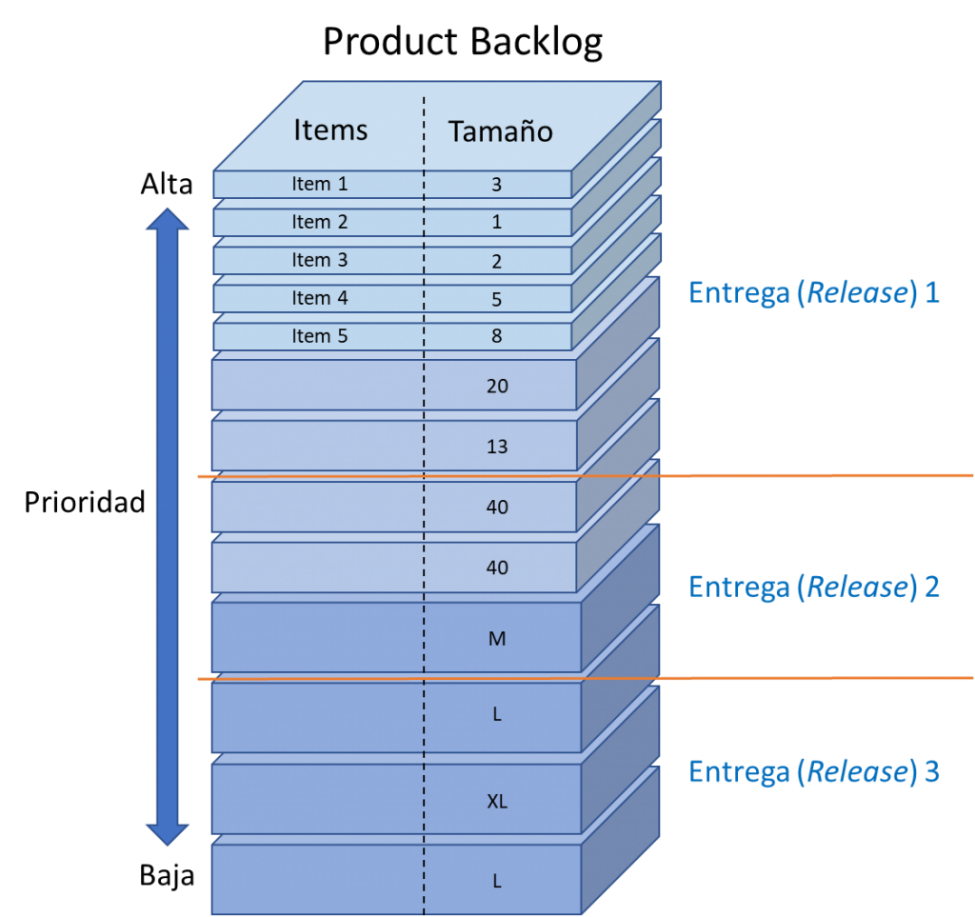
Se abarcará todo el ciclo de vida de DevOps y se elaborará un plan de acción para cada fase en el desarrollo del proyecto.

# Planeación

Para la gestión y planeación del proyecto no basaremos en una metodología ágil para hacer los procesos más rápidos. En este caso nos basaremos en la metodología Scrum para poder elaborar el plan de desarrollo.

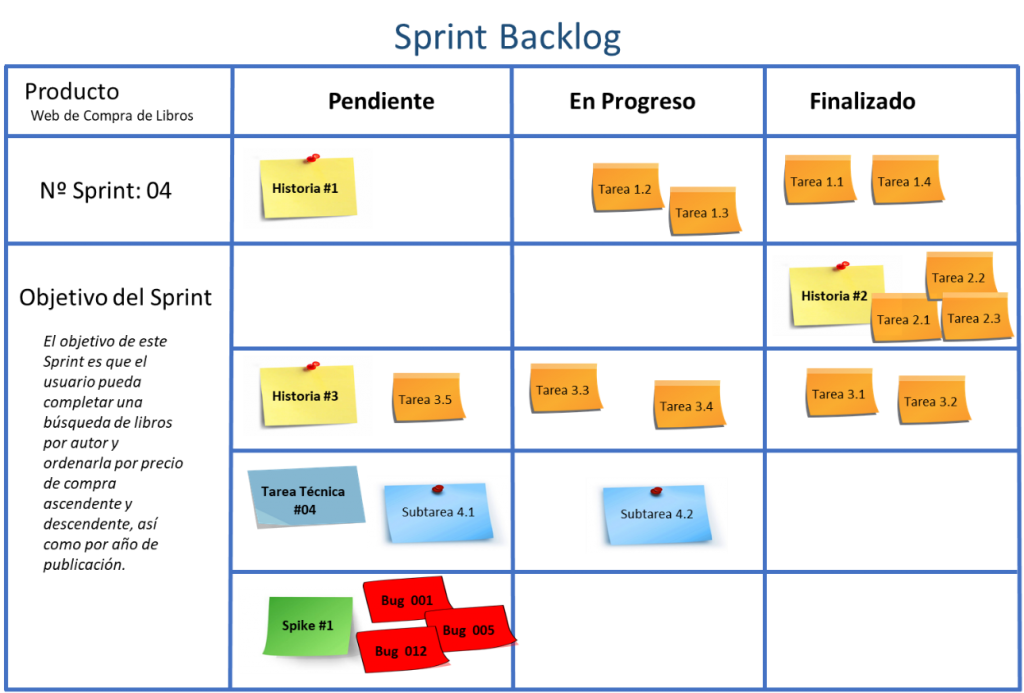


Primero crearemos nuestro Product baglog de acuerdo con el alcance definido para el desarrollo durante este cuatrimestre se enlistarán las actividades a realizar en los sprints, las actividades se clasificarán en orden de priorización de alto, medio y bajo iniciando los sprints con las actividades definidas como de alta prioridad.



Consecuentemente ya habiendo definido la prioridad de actividades se hará la planeación de los sprints de trabajo, para esto, haremos un sprint backlog donde por cada sprint se definirá un objetivo a cumplir, la fecha de entrega y las estrategias con las que se hará el código del proyecto.

Para los entregables definidos en los sprints deberemos contar con ciertos documentos que avalen o que ratifiquen lo que se estará codificando, en este caso lo primero que se hará son las historias de usuario para revisar el proceso que conllevara cada módulo, los diseños de las interfaces se harán aplicando los estándares para IU de Desing. Se definirá un plan de seguridad para resguardar la información y aportar una mejor calidad al producto. Se hará un plan de pruebas para cuando llegue el momento de la implementación los posibles problemas que puedan surgir ya se hayan corregido y pueda ser liberado sin problemas. Se utilizará Git hub como contenedor de versionamiento del código.



Al final de cada sprint se revisarán los resultados obtenidos con todo el equipo para detectar errores y corregirlos antes de pasar con el siguiente sprint.

## Alcance

La aplicación móvil Music Bar para la automatización del sistema; donde el usuario podrá registrarse e iniciar sesión para visualizar las promociones y eventos del bar.

|  |  |
| --- | --- |
| MB\_Login |  |
| MB\_Registro |  |

## Arquitectura

La arquitectura que usaremos para implementar este proyecto será la de microservicios, esta arquitectura nos permitirá separar todos los procesos, por lo cual, se llevará de la mano con el enfoque de DevOps donde empaquetaremos los módulos codificados en contenedores separados para poder llevarlo a producción de manera más eficiente, esto sumado a que la app estará alojada en un proveedor de nube evitara que la arquitectura consuma demasiados recursos estando instalada en un servidor físico, al mismo tiempo el proveedor de nube nos aportara la seguridad, escalabilidad, integridad y disponibilidad de datos. Además de tener ciertas ventajas al permitir actualizaciones más flexibles.

**Diagrama

Descripción generada automáticamente**

# Code

## Desarrollo

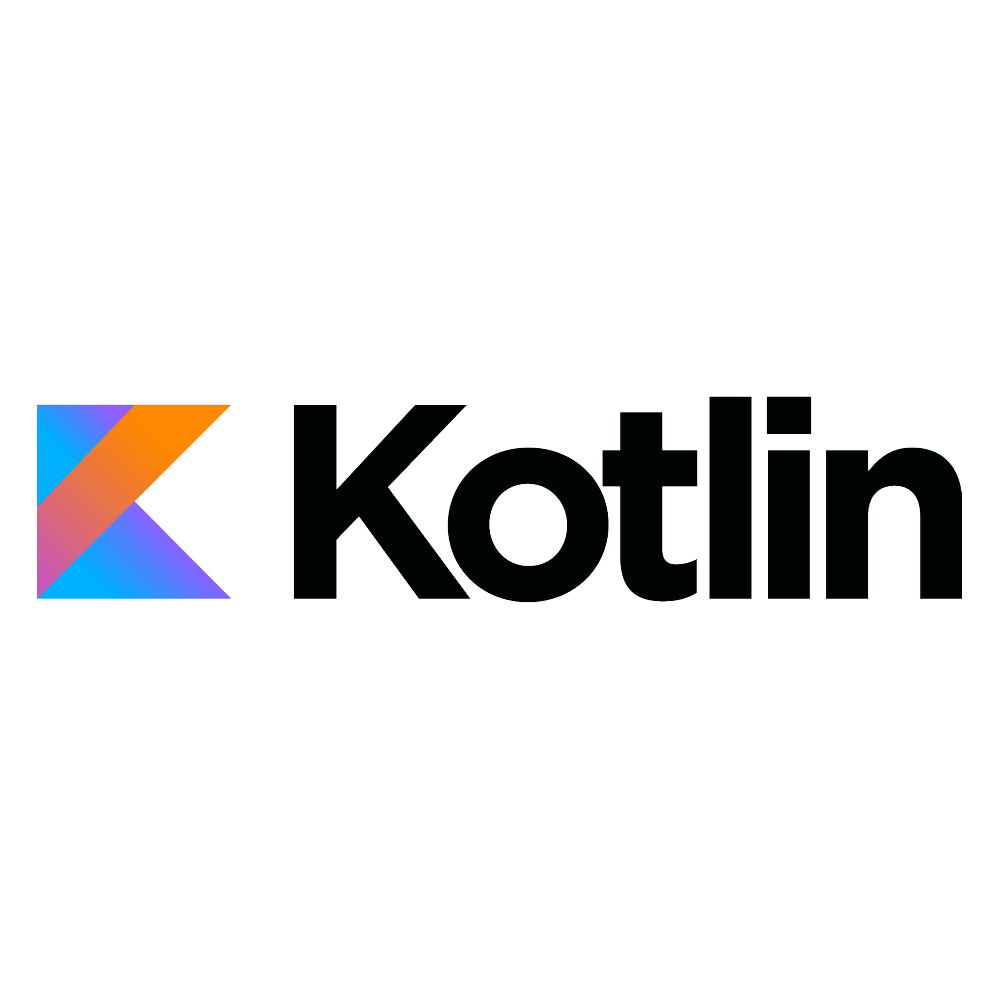
Para el desarrollo de nuestro proyecto utilizaremos varias herramientas para facilitar el trabajo y hacer un producto de mayor calidad y con mayor eficiencia.

Como frameworks de trabajo utilizaremos Android Studio Bumblebee versión 2021.1.1 para emular los resultados de la programación en un dispositivo móvil, Kotlin 1.6.20 para la construcción del código debido a su compatibilidad con Android studio, además de estar basado en el lenguaje de programación principal de Android, JAVA.



Kotlin nos aportará ventaja por la recién inclusión de Kotlin a Gradle (Herramienta que usaremos para la construcción del código) como lenguaje para escribir scripts de compilación, además, Kotlin tiene una capa multiplataforma para aplicaciones nativas, lo cual, nos permitirá adaptar nuestra versión web sin perder la experiencia de usuario.

La gran cantidad de integrantes por parte de la comunidad de desarrolladores quienes aportan información y experiencias valiosas hacia el uso de la aplicación serán de gran ayuda en la curva de aprendizaje de este framework de trabajo.



# Build

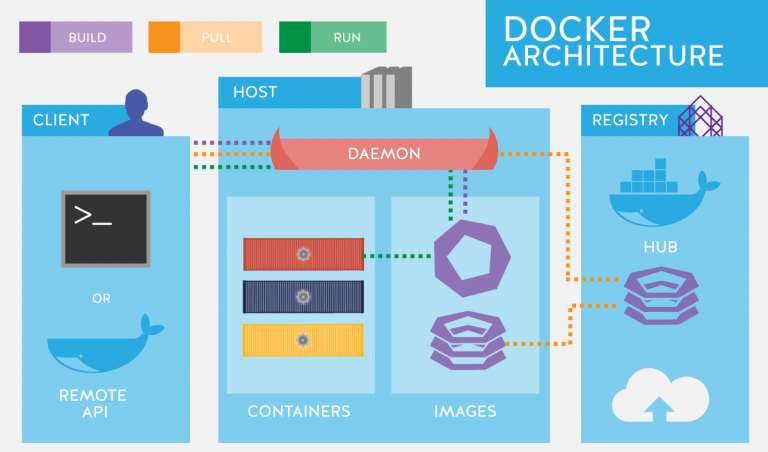
Una vez habiendo superado las primeras fases satisfactoriamente se creará la primera versión en plataforma móvil, estas versiones serán gestionadas desde GitHub.

La herramienta que usaremos para integrar la versión y ponerla ya en producción será Docker, el cual, nos permitirá empaquetar el código en contenedores para facilitar el despliegue de producción y los tiempos de ejecución.

La curva de aprendizaje de Docker no es muy difícil, por lo cual, debido a los tiempos ajustados se adaptará de buena forma al trabajo del equipo.



La ejecución del proyecto con Docker supondrá una ventaja para nuestro proyecto dado que Docker trabaja con imágenes creadas internamente y cada una de ellas trabaja por separado aportando librerías, extensiones y todo lo necesario para ejecutar la app.



# Test

Para la realización de pruebas se hará un plan donde en cada sprint terminado el código habrá sido sometido a un testeo para detectar errores antes de su aceptación. Esto nos permitirá construir de manera más eficiente y rápida nuestro proyecto.

El enfoque de DevOps nos permite hacer pruebas automatizadas del código antes de estar en producción siquiera, por lo cual, nos enfocaremos en un testeo de varios niveles: API, interfaz de usuario (IU) y unitarias.

* Unitarias: Con estas pruebas validaremos el correcto funcionamiento de cada componente creado en los sprints.
* Despliegue de ambiente QA: Haremos pruebas en un ambiente controlado y lo más parecido posible al ambiente de producción, por lo que, crear una virtualización del proyecto nos permitirá ejecutar este tipo de pruebas.
* API: Con la ayuda de NeoLoad automatizaremos las pruebas de las posibles APIs creadas para el proyecto un futuro.
* Pruebas de performance: Con estas pruebas mediremos la velocidad y el gasto de recursos consumidos por la aplicación, al mismo tiempo integraremos las pruebas de carga para medir la cantidad de usuarios que soportará la app.
* Pruebas de UI: Nos permitirán asegurar que la app cumpla con los requisitos funcionales y obtengamos estándares altos de calidad.

La herramienta que usaremos para la prueba del código será NeoLoad para medir el rendimiento del código, esta herramienta gracias a sus funciones específicas para la creación, mantenimiento y ejecución de las pruebas de manera más ágil nos permitirá ahorrar tiempo en el testeo por bloque de código.



# Release

Como contenedor de versionado usaremos Github donde todas las modificaciones o nuevas creaciones para el proyecto se guardarán y así podremos tener un historial controlado de la aplicación esto es una buena práctica en caso de ocurrir problemas con el proyecto ya estando en implementación o en caso de querer consultar una versión especifica.



Por cada sprint se construirá un bloque de código y se almacenará en el repositorio de GitHub para poder ser integrado más adelante en la implementación del producto.

# Deploy

En esta etapa evaluaremos la posibilidad de usar Gradle como herramienta de compilación para unir nuestro código y comenzar a hacer las pruebas.

Gradle nos ayudará a automatizar de mejor manera nuestro código y gracias a la compatibilidad con varios lenguajes de programación y el ambiente multiplataforma que ofrece será una herramienta de gran ayuda en el desarrollo móvil.

Un dibujo animado con letras

Descripción generada automáticamente con confianza media

Posteriormente se generará la APK para ejecutar en sistemas Android, que será accesible atreves de la página web del bar, posteriormente se alojará en la tienda de aplicaciones play store



# Operate

En la fase de operación se implementará el uso de Kubernetes gracias a su amplia gama de compatibilidad con herramientas de empaquetado y en este caso con Docker.

Kubernetes nos permitirá gestionar de manera más eficiente la ejecución del proyecto ya en producción al automatizar ciertos procesos que nos llevaría más tiempo configurar de manera manual.

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

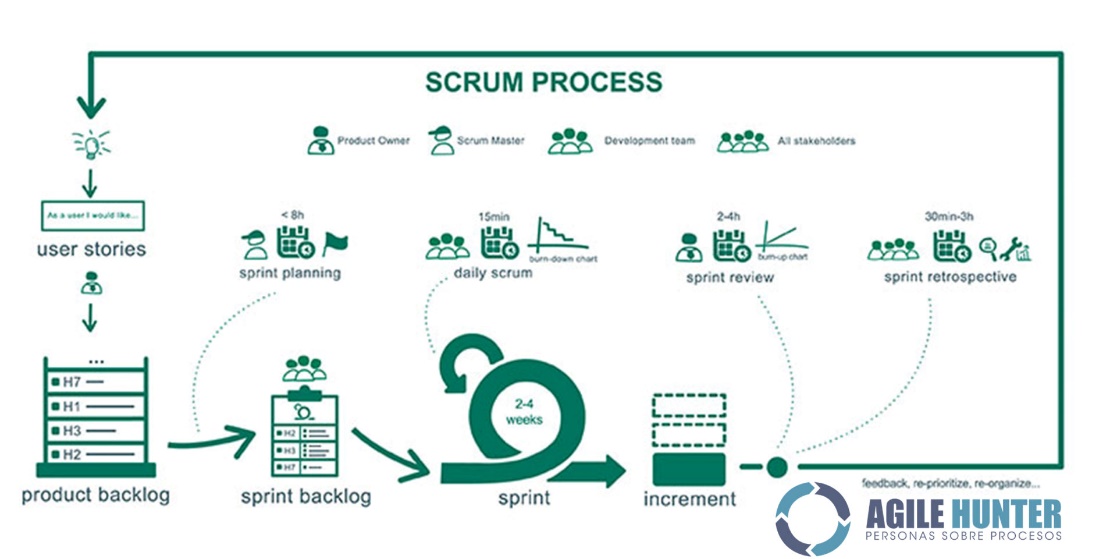
# Monitor

## En la fase de monitoreo

Las actividades de monitoreo y control del proyecto cobran importancia en la ejecución de este, ya que permiten realizar un seguimiento al trabajo del proyecto para compararlo con el trabajo planeado y así determinar si el proyecto está al día, atrasado o no se está cumpliendo el plan para alcanzar el objetivo del proyecto.

Se debe realizar el monitoreo acorde al plan de monitoreo y control del proyecto definido desde la planeación de este, este monitoreo arroja información clave del proyecto para determinar si es necesario realizar solicitudes de cambio o avanzar con el trabajo del proyecto



Para tener un mejor control en el desarrollo del proyecto nos basaremos en una metodología de gestión de proyectos “SCRUM”, puesto que al ser una metodología ágil nos va a permitir tener un mejor control en cada uno de los procesos que se llevan a cabo durante la realización de este.