



**IES AUGUSTO GONZÁLEZ DE
LINARES**
DEPARTAMENTO DE INFORMÁTICA

PROYECTO CHIKARA

**0492 - PROYECTO DE DESARROLLO DE
APLICACIONES MULTIPLATAFORMA**

CICLO FORMATIVO DE GRADO SUPERIOR

**DESARROLLO DE APLICACIONES
MULTIPLATAFORMA**

2023/2024

Díez de Paulino, Albano

Espinosa García, Daniel

Gutiérrez Valverde, Ramiro

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

Histórico de modificaciones

| Versión | Fecha | Hecho | Descripción |
|---------|------------|-------------------|-----------------------------------------------------|
| 1 | 02/04/2024 | Documento Inicial | Documentación de proyecto |
| 2 | 05/04/2024 | Segunda Entrega | Modelo de datos |
| 3 | 15/05/2024 | Tercera Entrega | Diseño de la interfaz |
| 4 | 21/05/2024 | Cuarta Entrega | Implementación |
| 5 | 31/05/2024 | Quinta Entrega | Puesta en Marcha, Plan de Pruebas y Plan de Empresa |
| 6 | 10/06/2024 | Entrega Final | Documentación de proyecto |

Licencia del documento



El presente documento se encuentra sujeto a la licencia Creative Commons BY 4.0 Internacional, si no dispone de una copia visite el siguiente sitio web

<https://creativecommons.org/licenses/by/4.0/>



ÍNDICE

| | | |
|--------|-----------------------------------|----|
| 1. | RESUMEN | 7 |
| 2. | PALABRAS CLAVE..... | 9 |
| 3. | INTRODUCCIÓN..... | 12 |
| 4. | OBJETIVOS..... | 14 |
| 4.1. | Requisitos funcionales | 14 |
| 4.2. | Requisitos no funcionales | 15 |
| 5. | ANÁLISIS DEL CONTEXTO | 18 |
| 5.1. | Análisis del contexto..... | 18 |
| 5.2. | Innovación | 19 |
| 6. | PLANIFICACIÓN | 21 |
| 6.1. | Diagrama de Gantt | 22 |
| 6.2. | Recursos técnicos y humanos | 26 |
| 6.3. | Control de Versiones | 26 |
| 7. | DISEÑO | 28 |
| 7.1. | Mockup | 28 |
| 7.2. | Casos de Uso..... | 30 |
| 7.3. | Modelos de datos | 31 |
| 7.3.1. | PostgresSQL | 31 |
| 7.3.2. | MongoDB | 32 |
| 7.4. | Estructura del proyecto | 32 |
| 7.4.1. | Arquitectura Back-end | 33 |
| 7.4.2. | Arquitectura Front-end | 37 |
| 7.5. | Despliegue | 38 |
| 7.5.1. | CI/CD | 41 |
| 8. | IMPLEMENTACIÓN | 45 |
| 8.1. | Frontend | 45 |
| 8.2. | Backend | 62 |



| | | |
|---------|-----------------------------------------|----|
| 9. | PUESTA EN MARCHA | 67 |
| 9.1. | Securización..... | 67 |
| 9.2. | Legalidad | 69 |
| 9.3. | De entorno DEV a PRO..... | 70 |
| 10. | PLAN DE PRUEBAS | 74 |
| 10.1. | Test Unitarios..... | 74 |
| 10.2. | Test de esfuerzo | 76 |
| 10.3. | Plan de Pruebas nivel Usuario | 78 |
| 10.4. | Beta Testers..... | 81 |
| 11. | PLAN DE EMPRESA..... | 83 |
| 11.1. | IDENTIFICACIÓN | 83 |
| 11.1.1. | Historia de la empresa..... | 83 |
| 11.1.2. | Cultura empresarial | 83 |
| 11.1.3. | Identificación del equipo | 87 |
| 11.1.4. | Descripción actividad empresarial | 88 |
| 11.2. | ANALISIS DE MERCADO | 89 |
| 11.2.1. | Análisis Macroentorno..... | 89 |
| 11.2.2. | Datos generales del sector..... | 90 |
| 11.2.3. | Clientes Potenciales | 92 |
| 11.2.4. | Plan de Marketing | 94 |
| 11.2.5. | DAFO..... | 95 |
| 11.3. | PLAN TECNICO-PRODUCTIVO..... | 95 |
| 11.3.1. | Servicios Ofrecidos | 95 |
| 11.3.2. | Tabla de Precios | 95 |
| 11.3.3. | Distribución | 96 |
| 11.4. | PLAN DE RECURSOS HUMANOS | 96 |
| 11.4.1. | Perfiles profesionales | 96 |
| 11.4.2. | Organigrama | 96 |

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|---------------------------------------------------------|------------|
| 11.4.3. Política Salarial | 96 |
| 11.4.4. Política de Teletrabajo | 97 |
| 11.4.5. Política acoso Laboral..... | 97 |
| 11.5. PLAN DE INVERSIONES | 98 |
| 11.5.1. Localización Sede | 98 |
| 11.5.2. Inmovilizado intangible. | 100 |
| 11.5.3. Inmovilizado Material..... | 100 |
| 11.5.4. Presupuesto Detallado de Gastos | 100 |
| 11.5.5. Resumen de Inversiones | 101 |
| 11.5.6. Fuentes de financiación | 101 |
| 11.6. OBLIGACIONES EMPRESARIALES | 102 |
| 11.6.1. Forma jurídica..... | 102 |
| 11.6.2. Ventajas | 102 |
| 11.6.3. Desventajas | 103 |
| 11.6.4. Pasos que rellenar para formar la empresa | 104 |
| 11.6.5. Obligaciones Laborales..... | 105 |
| 11.6.6. Obligaciones sobre prevención Laboral | 106 |
| 12. CONCLUSIONES..... | 109 |
| 13. BIBLIOGRAFÍA | 114 |
| 14. ÍNDICES | 116 |
| 14.1. Índice de Ilustraciones | 116 |
| ANEXOS | 119 |
| Anexo I Registro Horario..... | 119 |
| Anexo II Estatutos de Empresa | 120 |
| Anexo III Modelo 036..... | 122 |
| PREGUNTAS FRECUENTES..... | 125 |



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

1. ¿Por qué se ha usado un modelo de datos relacional y un modelo de datos no-relacional conjuntamente? 125
2. ¿Cómo se relacionan los modelos de datos usados? 125
3. ¿Por qué hemos usado el servicio cloud Azure de Microsoft sobre otras alternativas? 125
4. ¿Por qué usamos una arquitectura de microservicios en vez de una monolítica? 126
5. ¿El desarrollo móvil es solo para dispositivos con SO Android, tenemos planes para un desarrollo para SO iOS? ¿Qué lenguaje usaríamos? 126
6. ¿Si se deja planteado un desarrollo para múltiples dispositivos porque no hemos usado un framework multiplataforma? 126

RESUMEN

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

1. RESUMEN

Chikara es una aplicación multiplataforma que, en la fase actual del proyecto se concreta en una aplicación móvil para dispositivos Android, pero que puede ser implementada para otros sistemas operativos y plataformas.

Se trata de una red social que busca convertirse en un espacio de motivación personal en la que los usuarios pueden obtener inspiración de otros usuarios, o crear su propio contenido inspirador. En Chikara ponemos la fuerza que necesitas en un momento de flaqueza dentro de tu bolsillo. Además, podrás regalar fuerza a tus contactos ([nakamas](#)), creando contenido pensado para ellos.

Los [chiks](#), nombre que le hemos puesto a dicho contenido, se componen de imágenes y textos con gran carga emotiva para el usuario que, en conjunto, ofrecen un impulso en una determinada dirección.

A diferencia de otras redes sociales que se nutren de tu atención, drenando tu energía y consumiéndote entre caminos sinuosos, Chikara te proporciona foco, dirección e impulso. Puedes pasar el tiempo que quieras en chikara, pero ese tiempo se traducirá en motivación. Fuerza. Chikara.

PALABRAS

CLAVE

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

2. PALABRAS CLAVE

A continuación, se detalla una relación de los términos que se irán usando a lo largo de este documento:

- **Requisito funcional:** Requisito que define el comportamiento interno del desarrollo, como cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas.
- **Requisito no funcional:** Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema, como la estética o la seguridad.
- **Aplicación Multiplataforma:** Aplicación que puede ser ejecutada desde diferentes plataformas y/o Sistemas Operativos.
- **Base de datos:** Identifica dónde se encuentra alojada la información que utiliza el proyecto.
- **Front-end:** Interfaz visual del sistema que se desarrolle.
- **Back-end:** Operaciones y lógicas que realiza el sistema desarrollado entre la base de datos y el front-end.
- **API Rest:** Interfaz de programación de aplicaciones que permite la comunicación entre dispositivos.
- **Cloud o Cloud Computing:** Uso de una red de servidores remotos bajo demanda conectados a internet para procesar, almacenar o interconectar.
- **Microservicios:** Arquitectura del software que representa la separación de los servicios de forma independiente.
- **Red Social:** Estructura social compuesta por un conjunto de actores con una o más relaciones definidas entre ellos.
- **Nakama:** Palabra japonesa para referirse a amigo, que será utilizada en la aplicación para nombrar a los contactos.
- **Chikara:** Palabra japonesa que significa fuerza, utilizada para referirse al nombre de la aplicación



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

- **Chik:** Adaptación del término chikara que será utilizado para referirse a cada uno de los bloques de contenido que los usuarios crean a modo de motivación. Estará compuesto de uno o varios elementos. Pueden ser de texto o imágenes.
- **Chiks:** Plural de chik. Se refiere al conjunto de estos.

INTRODUCCIÓN



3. INTRODUCCIÓN

El ser humano es increíble, a todos los niveles. Nunca deja de sorprendernos con sus capacidades. Biológicamente no somos muy diferentes de la mayoría de los mamíferos que existen, pero a diferencia de ellos poseemos una mente superior.

En la mayoría de las ocasiones ésta trabaja a nuestro favor. Una herramienta capaz de encontrar soluciones a los problemas más diversos imaginables y, a la vez, de conseguir lo inimaginable en la búsqueda de la excelencia y la perfección. Pero, en ocasiones la mente deja de ser nuestra herramienta, nuestro aliado primordial, y se convierte en un obstáculo a superar. La falta de motivación, la pérdida de foco, impulso, dirección, como quiera llamarse se traduce en fallo, abandono, desidia, sensación de impotencia.

Nosotros en Café Con Palito vemos esa llama como la fuerza necesaria para llevar a buen término los proyectos vitales y profesionales. Esta aplicación es un brindis, un reconocimiento a esa fuerza y a su importancia en la historia de la humanidad.

Chikara y los chik.

Si hay una cultura que pueda destacar sobre el resto en su dedicación, tesón y respeto por la persistencia y el trabajo duro esa es la japonesa. Por eso el nombre de nuestra aplicación va a ser **CHIKARA**, la palabra japonesa para la fuerza. Su kanji, además, será el icono.

Un **CHIK** es un conjunto de imágenes y textos que contienen la fuerza, la energía que una persona necesita para superar cualquier bache y centrarse de nuevo en su objetivo. No es solamente motivación. Es tu motivación.

Crea un **CHIK** para cuando olvides por qué estás estudiando tan duro, entrenando, dejando de fumar, invirtiendo tanto tiempo en ese proyecto que no parece avanzar... Los esfuerzos parecen menores cuando se tiene la fuerza necesaria.

OBJETIVOS

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

4. OBJETIVOS

Los objetivos del presente punto son:

1. Definir exhaustivamente los requisitos deseables para el sistema que se pretende desarrollar.
2. Determinar el alcance definitivo del proyecto para adecuarlo a las horas de desarrollo contratadas.

Se definirán, por tanto, los requisitos que debe cumplir el sistema en su conjunto más amplio, sobrepasando los límites de tiempo de desarrollo del proyecto, especificando los detalles técnicos necesarios para llevarlos a cabo (fuentes u orígenes de datos, procesos de transformación, algoritmia, etc.) y, posteriormente, se etiquetarán conforme a su consideración dentro del proyecto como **[dentro de alcance]** o **[fuera del alcance]**.

Los requisitos **[dentro de alcance]** serán abordados dentro de los límites del proyecto actual y los **[fuera de alcance]** no se llevarán a cabo en el proyecto actual, pero se tendrán en cuenta en el diseño con el fin de permitir la escalabilidad del sistema.

En el supuesto de no contar con la suficiente información en relación con algún punto, se marcará como **[pendiente de análisis]**. Es posible que a lo largo del desarrollo del proyecto algunos o todos los elementos marcados con esta etiqueta se analicen en profundidad, lo que puede provocar su inclusión como **[dentro de alcance]**. Si no llegaran a analizarse, o una vez analizados no suponen un cambio en las prioridades del proyecto, estos puntos quedarán **[fuera de alcance]** y se contemplarían en futuras ampliaciones del proyecto.

4.1. Requisitos funcionales

En este apartado se van a determinar todos aquellos requisitos del sistema que son determinantes para la funcionalidad, que influyen en el mantenimiento, usabilidad o escalabilidad.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Aplicación Android
 - Sistema de login **[dentro de alcance]**.
 - Sistema de registro **[dentro de alcance]**.
 - Visualización de tus chiks **[dentro de alcance]**.
 - Buscador de chiks **[dentro de alcance]**.
 - Añadir nuevo chik **[dentro de alcance]**.
 - Eliminar y modificar chiks **[dentro de alcance]**.
- Api Rest
 - Comunicación PostgreSQL **[dentro de alcance]**.
 - Comunicación MongoDB **[dentro de alcance]**.
 - Securización (JWT) **[dentro de alcance]**.
 - Comunicación Blob Storage **[dentro de alcance]**.
- Bases de datos
 - Sistema gestor de bases de datos PostgreSQL **[dentro de alcance]**.
 - Sistema gestor de bases de datos MongoDB **[dentro de alcance]**.
- Aplicación de escritorio
 - Sistema de login **[fuera de alcance]**.
 - Sistema de registro **[fuera de alcance]**.
 - Visualización de tus chiks **[fuera de alcance]**.
 - Buscador de chiks **[fuera de alcance]**.
 - Añadir nuevo chik **[fuera de alcance]**.
 - Eliminar y modificar chiks **[fuera de alcance]**.

4.2.Requisitos no funcionales

En este apartado se van a definir todos aquellos requisitos del sistema que no son determinantes para la funcionalidad, pero que influyen en el mantenimiento, usabilidad, escalabilidad, como puedan ser arquitecturas, necesidades de recarga de la información, definición de perfiles, etc.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Aplicación Android
 - Sistema de nakamas [**dentro de alcance**].
 - Sistema de recuperación de contraseña [**dentro de alcance**].
 - Sistema de invitación [**fuera de alcance**].
 - Sistema de likes [**fuera de alcance**].
 - Etiquetado [**fuera de alcance**].
 - Comentarios [**fuera de alcance**].
 - Foto de usuario [**pendiente de análisis**].
- Api Rest
 - Sistema de notificación de nuevas conexiones por correo [**dentro de alcance**].
 - Lógica de recuperación de contraseña [**dentro de alcance**].
 - Lógica de nakamas [**fuera de alcance**].
 - Lógica de invitaciones [**fuera de alcance**].
 - Lógica de likes [**dentro de alcance**].
 - Lógica de etiquetado [**dentro de alcance**].
 - Lógica de comentarios [**dentro de alcance**].
 - Lógica foto de usuario [**pendiente de análisis**].
- Aplicación de escritorio
 - Sistema de nakamas [**fuera de alcance**].
 - Sistema de recuperación de contraseña [**fuera de alcance**].
 - Sistema de invitación [**fuera de alcance**].
 - Sistema de likes [**fuera de alcance**].
 - Etiquetado [**fuera de alcance**].
 - Comentarios [**fuera de alcance**].
 - Foto de usuario [**fuera de alcance**].

ANÁLISIS DEL CONTEXTO

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

5. ANÁLISIS DEL CONTEXTO

5.1. Análisis del contexto

En la actualidad, haciendo un repaso de las principales redes sociales del mercado, no encontramos ninguna que realice la actividad que proponemos desde Café con Palito. Si bien es cierto que algunas aplicaciones existentes, como Facebook o Instagram de Meta o TikTok de ByteDance, ofrecen funcionalidades que pueden albergar contenido similar al que nosotros desarrollamos, nunca es exclusivo, y desaparece, no consigue destacar entre el resto de contenido disminuyendo o anulando su eficacia.

Facebook, Instagram y TikTok, en un abanico de edades que abarcan todas las franjas posibles, atraen a un tipo de usuario que llega a consumir contenido variado, la mayoría pasivo, y a crear contenido para que otros consuman. Se basa en un sistema de recompensas y de algoritmos que ofrecen al usuario contenido dirigido, utilizando datos de este, que son recogidos por la propia aplicación o comprados por la empresa que las despliega. Aquí no hay más interés que el de mantener al usuario conectado y consumiendo anuncios, ya que se trata de aplicaciones gratuitas, con un alto mantenimiento y consumo de recursos, que han de responder ante su accionariado.

Un análisis DAFO del proyecto podría resumirse de la siguiente manera:

Debilidades:

- El mercado está sobresaturado de herramientas de autoayuda, redes sociales y aplicaciones en general, que disponen de un presupuesto mucho mayor que el nuestro, por lo que puede resultar muy complicado hacerse un hueco.
- La aplicación adquiere atractivo a medida que más y más usuarios se la descargan y la utilizan, por lo que en un principio puede no resultar llamativa.

Amenazas:

- Alguna empresa con más poder podría copiar la idea y eclipsar a Chikara dejándola en el olvido o como una mera anécdota.
- Al tratarse de una empresa de desarrollo incipiente, el usuario puede desconfiar de nosotros como desarrolladores y de Chikara como producto.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fortalezas:

- Se trata de una red social que realmente nace con el propósito de aportar valor y lo hace quitando todo lo que sobra, centrándose en un aspecto concreto: la motivación.
- Es una aplicación creada para ser accesible desde cualquier dispositivo o plataforma.
- Permite a los usuarios mantenerse en el anonimato decidiendo si hacer sus publicaciones privadas o no, y utilizarla para el único propósito de inspirarse.

Oportunidades:

- La motivación personal es un tema crucial en la época en la que estamos viviendo. En un mundo diseñado para distraernos y hacernos consumir pasivamente, algo que nos ayuda a enfocarnos y nos ofrece un empujón en un momento de necesidad puede ser un gran aliado.
- Hay personas con más o menos destreza para crear elementos audiovisuales capaces de conmover y llamar a la acción de los demás, pero todos se benefician de un chik de calidad. No hay por qué utilizar exclusivamente elementos propios para que el propósito tenga éxito.

5.2. Innovación

Nuestro principal cometido, a diferencia de otras plataformas es motivar. Por lo tanto, nuestros usuarios tienen el aliciente de encontrar una fuente de motivación cuando acceden a Chikara, y no solo la necesidad de conectar y compartir contenido o consumirlo, aunque también puedan. Chikara es una red social con una función bien diferenciada.

La experiencia del usuario es siempre diferente porque cada persona acude a Chikara con sus propias necesidades. El amplio margen de personalización que eso ofrece nos aporta una experiencia de usuario única y valiosa que puede marcar la diferencia.

PLANIFICACIÓN

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

6. PLANIFICACIÓN

Las horas efectivas de trabajo para el alcance actual de proyecto han sido definidas por el equipo de desarrollo en base a lo establecido en el boletín oficial del estado (BOE), “*Real Decreto 405/2023, de 29 de mayo, por el que se actualizan los títulos de la formación profesional del sistema educativo de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y Técnico Superior en Desarrollo de Aplicaciones Web, de la familia profesional Informática y Comunicaciones, y se fijan sus enseñanzas mínimas.*”, el cual establece una duración de 25 horas para el módulo “*0492 - proyecto de desarrollo de aplicaciones multiplataforma*” que se desarrollan en el último trimestre del segundo curso del ciclo formativo.

Dichas horas se dividen en 4 grupos diferenciados con diferente número de horas establecidas, aunque queda a libre asignación por cada desarrollador mientras no exceda un 15% el máximo de horas de cada grupo, a continuación, se detalla dichos máximos por desarrollador.

- Reuniones internas y externas: 20h
- Formación técnica: 30h
- Desarrollo e Investigación: 70h
- Pruebas: 10h
- Documentación: 20h
- Presentaciones: 2h
- **TOTAL: 152h**

Para el control de tareas y horas usamos el software web “*Jira*” de *Atlassian*, ya que este nos proporciona muchas herramientas para el desarrollo como, por ejemplo, un registro horario por tareas, un cronograma para ver qué tareas tienen más prioridad, o un Kanban para saber en qué estado se encuentran.



6.1. Diagrama de Gantt

Una vez determinado el número de horas para el alcance actual del proyecto, definimos la distribución en días desde el 06 de marzo de 2024 hasta el 16 de junio de 2024, un total de 102 días, es decir cada desarrollador debe como mínimo emplear 1h y 30min diarios, si no es así, dichas horas deben ser compensadas en los días posteriores, previos al último día del proyecto.

Para ser más eficaces en el desarrollo se utiliza una metodología ágil de trabajo conocida como “**SCRUM**”, por la cual se establecen periodos cortos durante el desarrollo denominados “**SPRINTS**”, en nuestro caso de 2 semanas, al final de los cuales se debe presentar avances en el desarrollo.

A continuación, se muestra un diagrama de Gantt con dichos periodos y lo que se tiene que desarrollar en ellos.



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

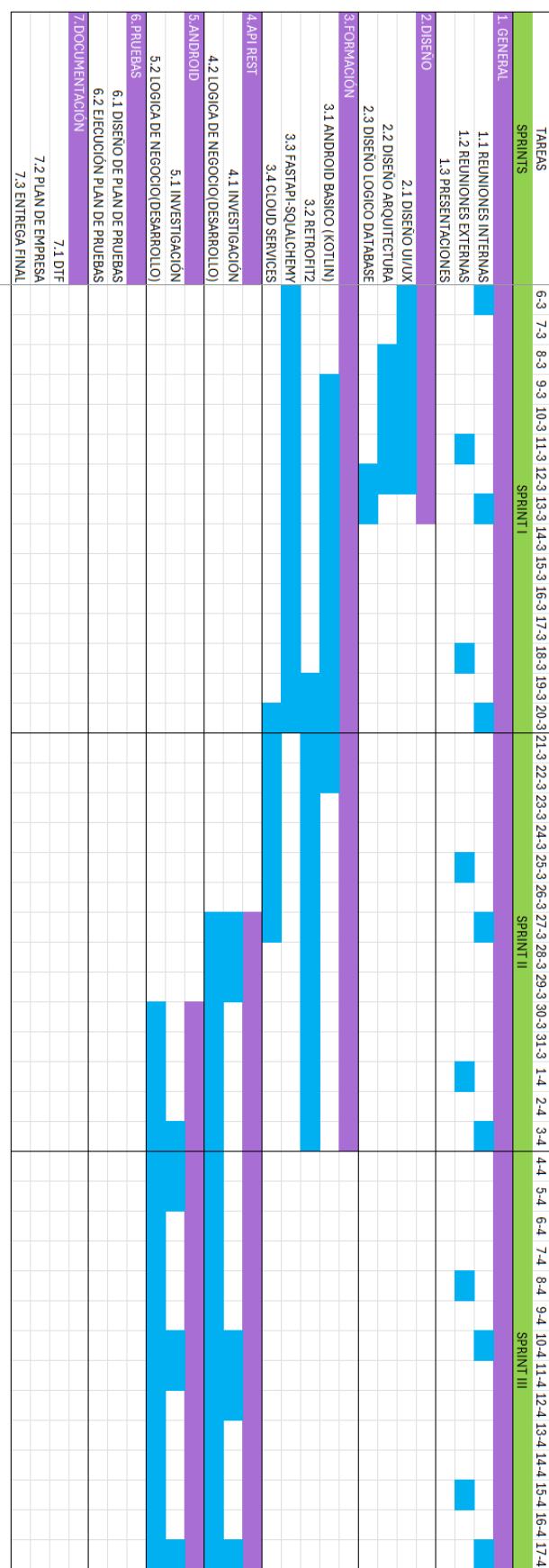


Diagrama 1 - Gantt PI



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

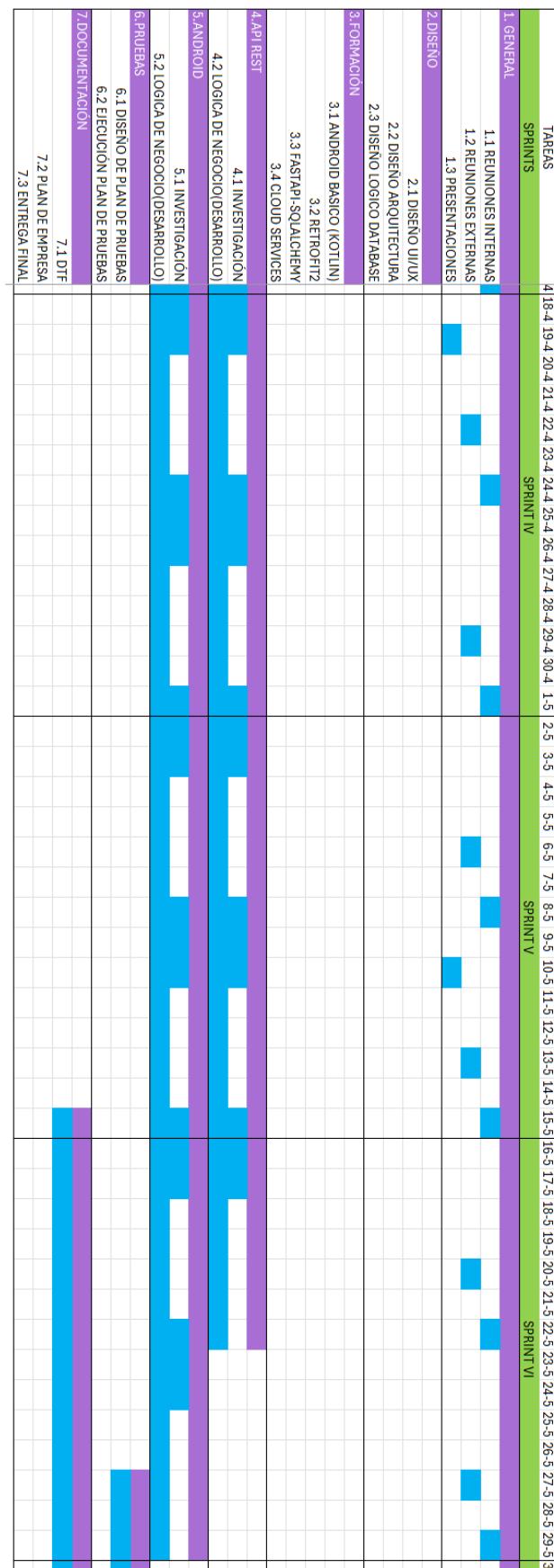


Diagrama 2 - Gantt P2



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

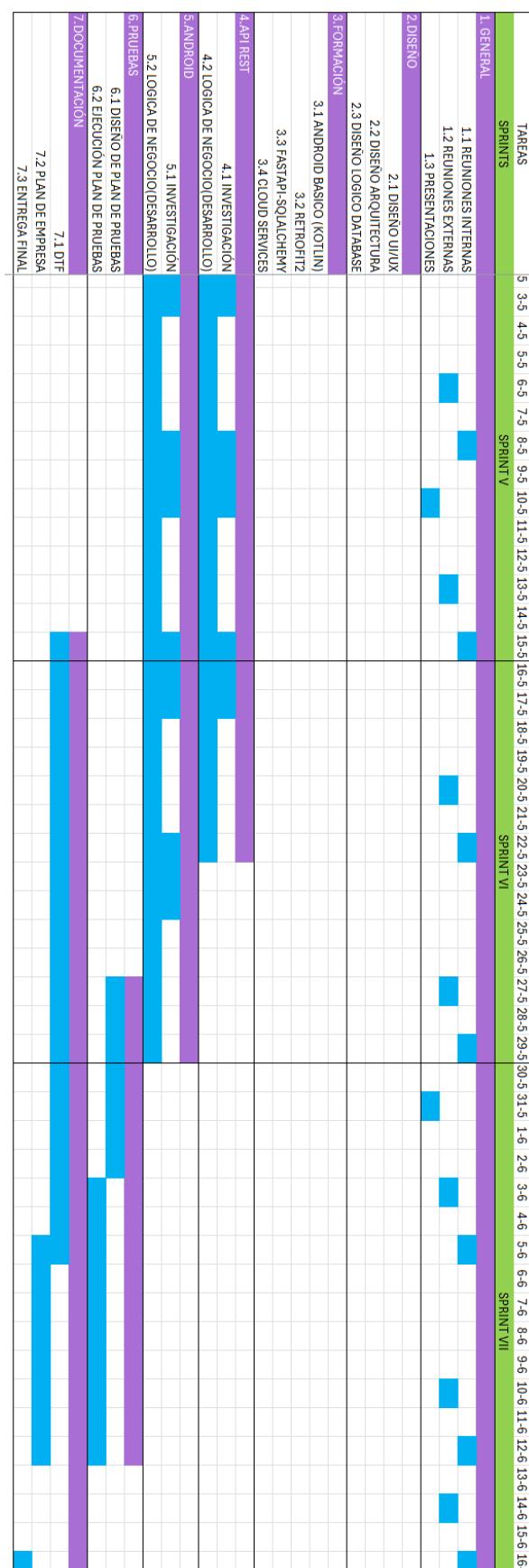


Diagrama 3 - Gantt P3

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

6.2. Recursos técnicos y humanos

Para la primera fase del proyecto se requieren 3 desarrolladores multiplataforma, que trabajen en todos los bloques.

Además, se requiere una cuenta en Azure para el despliegue de los entornos de desarrollo/producción y 3 equipos que cuenten con el siguiente software:

- Intérprete de Python 3.11.6 o superior.
- IDE para Python, preferiblemente VSCode o PyCharm.
- Android SDK.
- Compilador e intérprete de Kotlin.
- Android Studio.
- Docker.
- GIT.
- Jira.

6.3. Control de Versiones

El uso de un control de versiones es fundamental para gestionar proyectos de manera eficiente. Permite llevar un historial detallado de cambios y gestionar diferentes ramas de desarrollo, facilita la colaboración entre múltiples desarrolladores y proporciona mecanismos de respaldo y recuperación. Estas capacidades nos aseguran que podemos trabajar de manera ordenada y segura, con la capacidad de revertir errores y entender la evolución del proyecto a lo largo del tiempo.

Para dicho control hemos usado Git, es una herramienta ampliamente adoptada por sus ventajas como sistema distribuido, su rapidez en operaciones locales, y su flexibilidad para adaptarse a diversos flujos de trabajo. Su comunidad activa, junto con un ecosistema robusto de herramientas y plataformas como GitHub y GitLab, facilita la colaboración y la integración en entornos de desarrollo existentes. Además, Git ofrece una alta seguridad mediante la verificación de integridad de los datos, convirtiéndolo en una opción ideal para gestionar proyectos de software.

DISEÑO

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

7. DISEÑO

El diseño es la fase más fundamental del desarrollo, ya que lo que se plantea en esta fase condiciona al resto. Nuestro diseño se basa en nuestra competencia con ligeras modificaciones, ya que estos llevan años en el mercado y tienen un equipo muy amplio de diseñadores, por lo que saben lo que mejor funciona en una red social.

Nuestro primer paso fue diseñar un [mockup](#) y unos casos de uso, para luego determinar toda la [infraestructura](#) que vamos a necesitar.

7.1. Mockup

Los mockups son una herramienta invaluable en el proceso de diseño, permitiendo una visualización clara y realista de los diseños antes de su implementación final. Al utilizar mockups de manera efectiva, los diseñadores pueden obtener retroalimentación temprana, ahorrar tiempo y recursos, y garantizar la satisfacción del cliente. A continuación, se muestra el mockup de la aplicación móvil de chikara.

La app debe contener una pantalla de carga que sirva para comprobar que el servidor api rest está disponible, se pasa a la pantalla de login si hay conexión. Si no la hay se muestra un mensaje de fallo de conexión. Tras pasar el login, la navegación múltiple desde un menú inferior nos permite acceder a cualquier información en menos de dos clicks ([pulsaciones](#)).



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

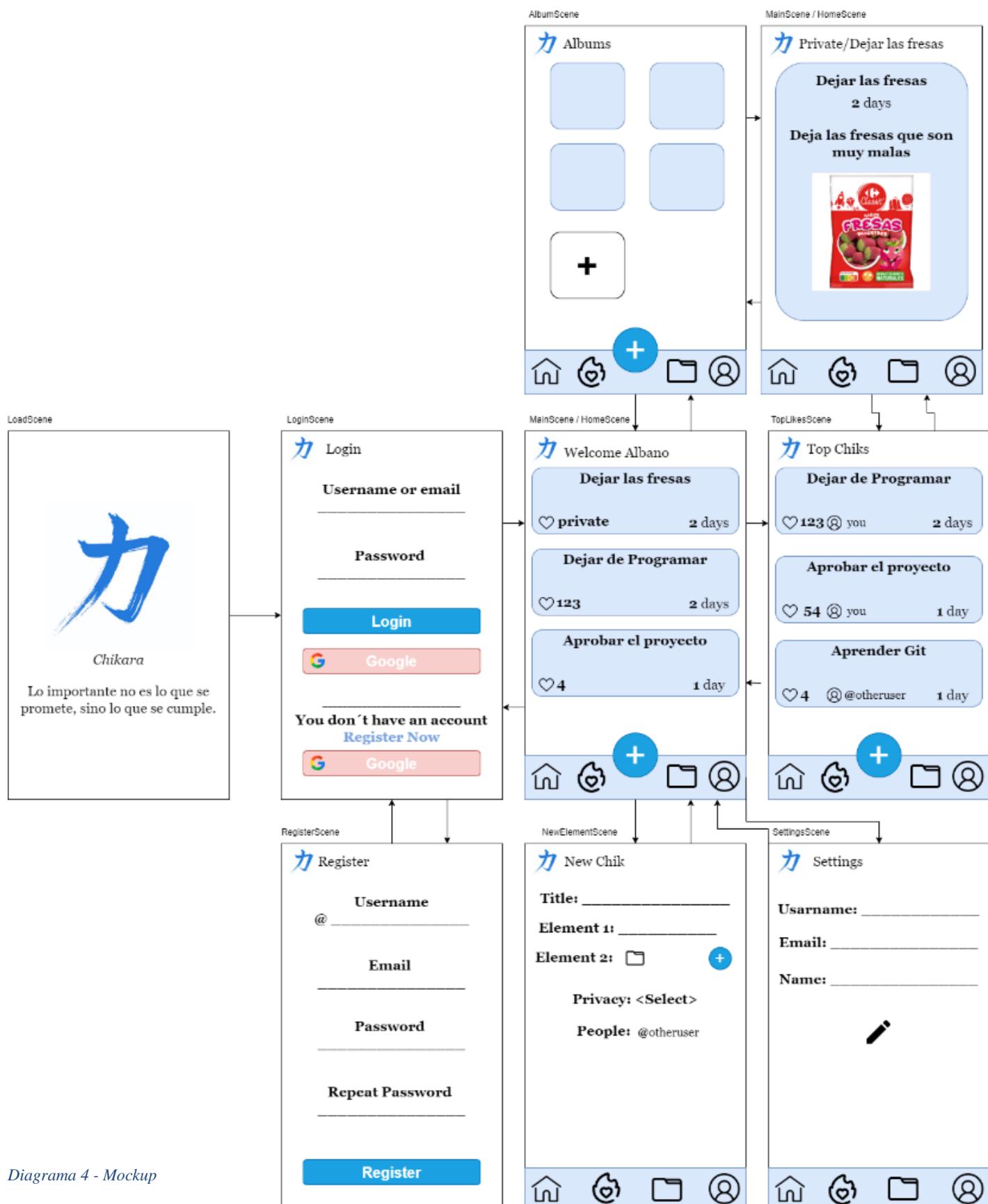


Diagrama 4 - Mockup



7.2.Casos de Uso

Los diagramas de uso son una herramienta esencial en el desarrollo de software, permitiendo una comprensión clara y detallada de cómo interactúan los usuarios con un sistema. Al utilizar diagramas de uso de manera efectiva, los equipos de desarrollo pueden clarificar requisitos, detectar problemas de diseño y diseñar interfaces de usuario intuitivas.

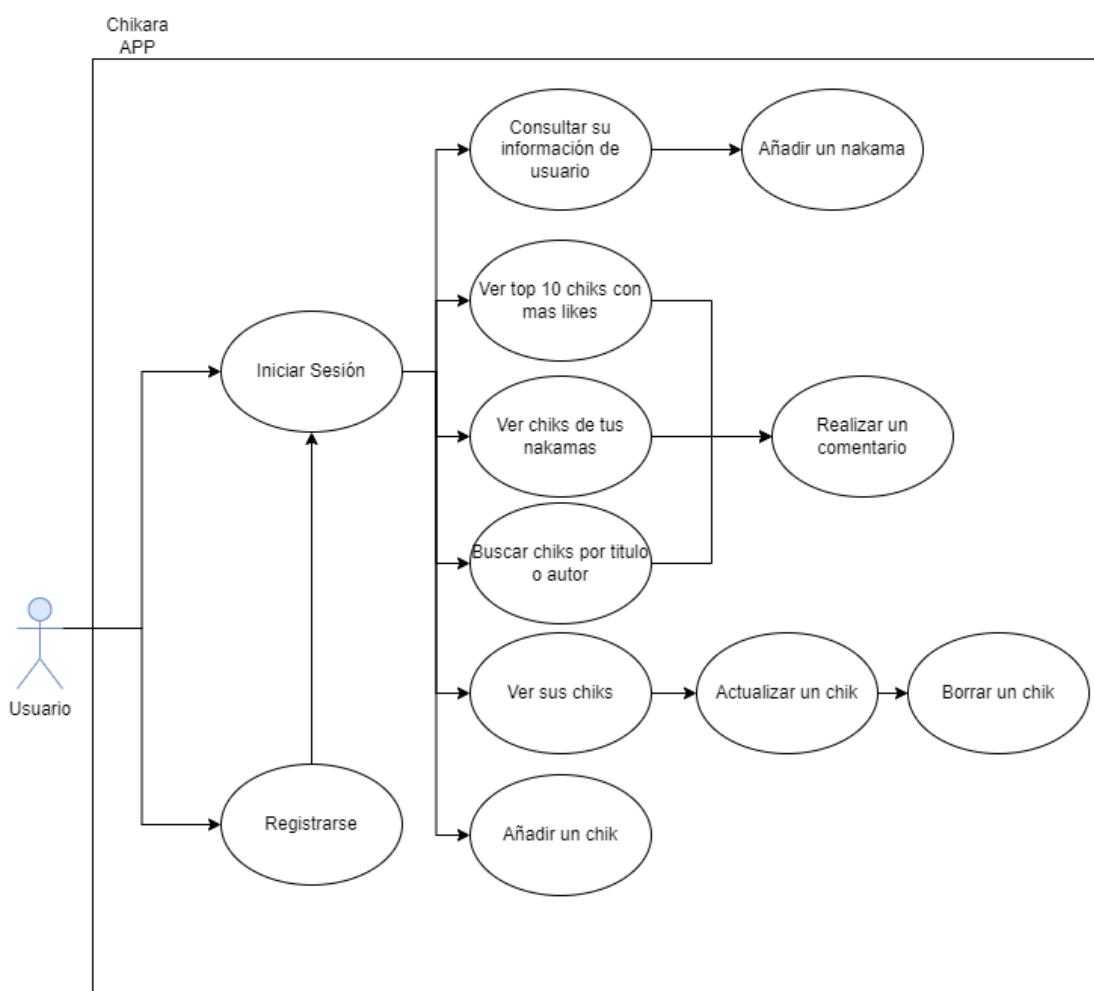


Diagrama 5 - Diagrama de Uso



7.3. Modelos de datos

El modelado de datos es una práctica esencial en el desarrollo de software que permite definir la estructura y las relaciones entre los datos utilizados por una aplicación. La utilización de modelos de datos de manera efectiva permite a los equipos de desarrollo clarificar la estructura de los datos, detectar problemas de diseño y crear una base sólida para el desarrollo de la base de datos.

En nuestro caso se optó por una aplicación con dos sistemas de almacenamiento de datos, uno relacional para almacenar la información de los usuarios y otro no-relacional para almacenar la información de los [chiks](#), relacionando ambos por la [UUID](#) del usuario.

7.3.1. PostgreSQL

El primer paso fue decidir qué información del usuario íbamos a guardar. Decidimos que íbamos a guardar los datos básicos del usuario, los dispositivos con los que ha iniciado sesión, un registro de inicio de sesiones, y qué [nakamas](#) tiene cada usuario. Para almacenar toda esa información diseñamos el siguiente esquema.

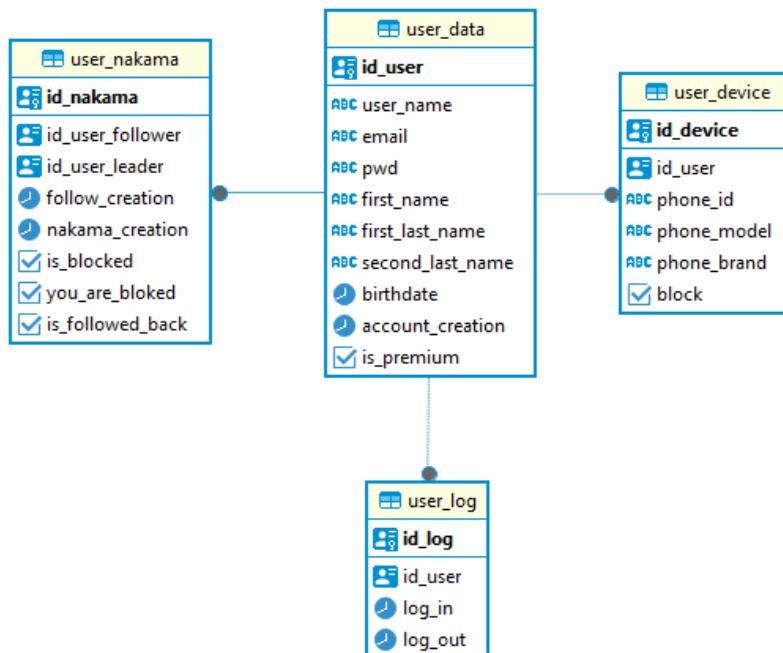


Diagrama 6 - Modelo Relacional



7.3.2. MongoDB

Toda la información referente a los **chiks** se almacena en documentos de **Mongo** con estructura **JSON**. A continuación, se muestra un ejemplo de **JSON** almacenado.

```
● ● ●
1  {
2    "_id": "1cad1ea5-b411-4895-812d-f5c3bf696519",
3    "title": "Prueba Chik",
4    "author": "156844a7-2be5-4208-b46d-7794fe698034",
5    "date": "2024-03-21",
6    "likes": 12,
7    "isprivate":true,
8    "content": [
9      {
10        "position":1,
11        "value":"Dejar las chuches",
12        "type":"TYPE_TEXT"
13      },
14      {
15        "position":2,
16        "value":"https://www.giantfreakinrobot.com/wp-content/uploads/2022/06/hellotherethumb.jpg",
17        "type":"TYPE_IMG"
18      }
19    ],
20    "comments": [
21      {
22        "user": "c3b63b33-281c-403f-8129-0ea95e56682f",
23        "comment": "Que chulo",
24        "date": "2024-03-21"
25      },
26      {
27        "user": "c28a49c1-63ae-4944-81d9-ba58f60741fa",
28        "comment": "Que guay",
29        "date": "2024-03-21"
30      }
31    ],
32    "mentions": ["c3b63b33-281c-403f-8129-0ea95e56682f", "c28a49c1-63ae-4944-81d9-ba58f60741fa", "1cad1ea5-b411-4895-812d-f5c3bf696519"]
33  }
```

Código 1 - Chik JSON

7.4. Estructura del proyecto

Dado que el proyecto es multiplataforma, hemos considerado esta característica para plantear su estructura. Hemos decidido desarrollar un proyecto cliente-servidor para lo cual hay que desarrollar una API genérica que pueda ser utilizada por múltiples clientes en diferentes plataformas. A continuación, se presenta un esquema general que se detallará en los apartados **0 Arquitectura Back-end**, **7.4.2 Arquitectura Front-end** y **7.5 Despliegue**. Es importante destacar que este no es un esquema UML, ya que la complejidad del proyecto, con su gran número de clases, lo haría difícil de entender.

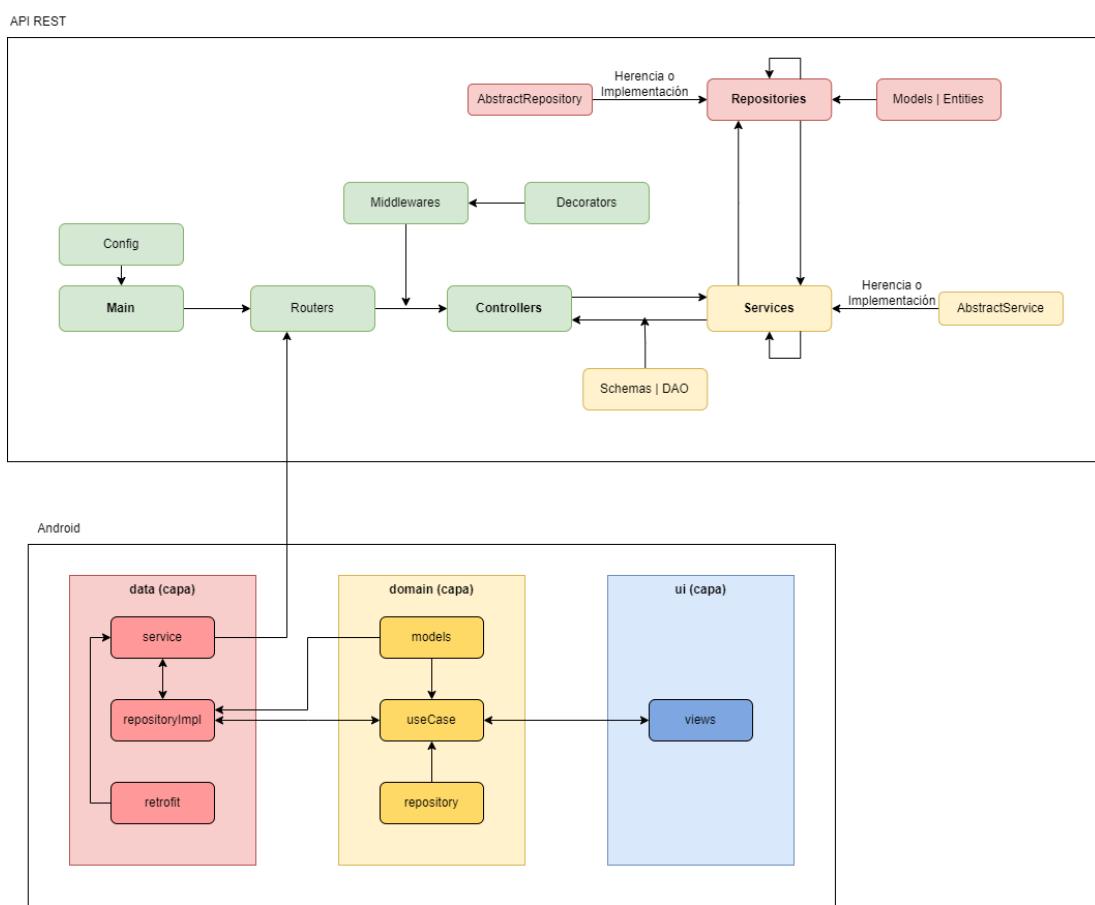


Diagrama 7 - Estructura Chikara

7.4.1. Arquitectura Back-end

El backend está desarrollado con Python, ya que cuenta con muchas librerías para programar de forma sencilla toda la lógica de negocio. Las principales dependencias son FastAPI, SQLAlchemy, PyMongo y Azure Services. La Arquitectura de clases que hemos seguido es la de Controller-Service-Repository, en la que cada elemento tiene una funcionalidad distinta, facilitando el mantenimiento y escalabilidad de la API Rest, que se detalla a continuación.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **MAIN**

- **Propósito:** Punto de inicio del servidor.
- **Responsabilidad:** Configurar y lanzar el servidor. Suele incluir la configuración inicial, como la carga de configuraciones, la conexión a la base de datos y la configuración de rutas y middleware.
- **Ejemplo:** En una aplicación FastAPI, aquí es donde se define la instancia de FastAPI y se ejecuta el servidor.

- **CONFIG**

- **Propósito:** Configuración de la aplicación.
- **Responsabilidad:** Gestionar todas las configuraciones necesarias para la aplicación, como configuraciones de la base de datos, variables de entorno, parámetros de configuración, etc.
- **Ejemplo:** Archivos como config.py en una aplicación Python, donde se definen variables como DATABASE_URL, SECRET_KEY, etc.

- **DECORATORS**

- **Propósito:** Funciones adicionales aplicadas a otras funciones o métodos.
- **Responsabilidad:** Añadir funcionalidad extra a las funciones o métodos, como autenticación, autorización, registro de logs, etc.
- **Ejemplo:** Decoradores que validan tokens de autenticación antes de permitir el acceso a un endpoint.

- **MIDDLEWARES**

- **Propósito:** Procesamiento adicional de solicitudes y respuestas.
- **Responsabilidad:** Interceptar las solicitudes antes de llegar a los controladores y/o interceptar las respuestas antes de enviarlas al cliente. Utilizados para tareas como la gestión de CORS, el manejo de sesiones, el registro de solicitudes, etc.
- **Ejemplo:** Middleware para gestionar CORS o registrar el tiempo de procesamiento de cada solicitud.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **ROUTERS**

- **Propósito:** Definición de rutas de la API.
- **Responsabilidad:** Encapsular y definir las rutas de la API. Organizar las rutas en diferentes módulos para mantener el código modular y limpio.
- **Ejemplo:** Definir routers en diferentes archivos para incluirlos en la aplicación principal.

- **CONTROLLERS**

- **Propósito:** Controladores de la lógica de negocio.
- **Responsabilidad:** Gestionar las solicitudes entrantes, llamar a los servicios necesarios y devolver las respuestas adecuadas. Actúan como una capa intermedia entre las rutas y los servicios.
- **Ejemplo:** Clases que reciben solicitudes HTTP, llaman a los servicios y devuelven las respuestas.

- **SERVICES**

- **Propósito:** Implementación de la lógica de negocio.
- **Responsabilidad:** Contener la lógica de negocio de la aplicación. Interactuar con los repositorios para obtener, procesar y devolver datos según las necesidades de la aplicación.
- **Ejemplo:** Métodos que implementan la lógica de negocio, como `create_user`, `add_user`, etc., que interactúan con los repositorios para realizar operaciones en la base de datos.

- **ABSTRACTSERVICE**

- **Propósito:** Definición de interfaces de servicio.
- **Responsabilidad:** Definir los métodos básicos que deben implementar los services. Garantiza que todas las implementaciones de servicios sigan un conjunto estándar de métodos.
- **Ejemplo:** Clase base con métodos abstractos que deben ser implementados por todas las clases de servicios.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **SCHEMAS**

- **Propósito:** Definición de esquemas.
- **Responsabilidad:** Definir la estructura de los datos que se envían y reciben a través de la API.
- **Ejemplo:** Modelos de Pydantic para validar y serializar datos de entrada/salida.

- **REPOSITORIES**

- **Propósito:** Acceso a los datos.
- **Responsabilidad:** Contener la lógica para acceder a los datos almacenados en la base de datos. Realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y otras consultas específicas.
- **Ejemplo:** Clases que contienen métodos para realizar operaciones en la base de datos, como UserRepository con métodos get_user_by_id, create_user, etc.

- **ABSTRACTREPOSITORY**

- **Propósito:** Definición de interfaces de repositorio.
- **Responsabilidad:** Definir los métodos que los repositorios deben implementar. Garantiza que todas las implementaciones de repositorios sigan un conjunto estándar de métodos.
- **Ejemplo:** Clase base con métodos abstractos que deben ser implementados por todos los repositorios.

- **MODELS | ENTITIES**

- **Propósito:** Definición de los modelos de datos.
- **Responsabilidad:** Representar la estructura y relaciones de las tablas de la base de datos.
- **Ejemplo:** Clases SQLAlchemy y Pydantic que definen los modelos de datos, como User, Chik, etc.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **INTERACCIONES ENTRE COMPONENTES**

- Main inicializa la configuración ([Config](#)) y define la instancia de la aplicación.
- Routers manejan las rutas de la API y delegan la lógica a los Controllers.
- Controllers llaman a los Services para realizar la lógica de negocio.
- Services interactúan con los Repositories para realizar operaciones en la base de datos.
- Repositories utilizan Models | Entities para representar y manipular los datos.
- AbstractRepository y AbstractService definen interfaces que aseguran consistencia en las implementaciones de Repositories y Services.

7.4.2. Arquitectura Front-end

El frontend ha sido desarrollado en Kotlin, debido a que es el que más soporte tiene para los dispositivos Android. Además, hemos usado múltiples dependencias. La principal es Retrofit que nos permite realizar consultas http a al backend ([API rest](#)), para tener una buena arquitectura y que el proyecto sea fácilmente escalable hemos usado las siguientes categorías de clases por capa.

- **CAPA “DATA”**

Esta capa es responsable de manejar los datos de la aplicación, ya sea desde una base de datos local, una API remota, o cualquier otra fuente de datos. Incluye los siguientes componentes:

- **service:** Esta es la interfaz que define las operaciones que se pueden realizar para obtener o enviar datos. Es la capa más externa que interactúa con los repositorios.
- **repositoryImpl:** Implementación del repositorio, que se encarga de interactuar con las fuentes de datos ([como bases de datos locales o servicios web](#)). En este caso, usa Retrofit para las llamadas a servicios web.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **retrofit:** Una biblioteca de cliente HTTP que facilita la comunicación con servicios web RESTful. Es utilizado por repositoryImpl para hacer las peticiones de red.

- **CAPA “DOMAIN”**

Esta capa contiene la lógica de negocio de la aplicación y actúa como un intermediario entre la capa de datos y la capa de UI. Incluye:

- **models:** Define las estructuras de datos que se usan en la aplicación, estos modelos son los que la capa de dominio maneja y manipula.
- **useCase:** Representa un caso de uso específico de la aplicación. Es una clase que encapsula una funcionalidad particular de la aplicación, como "obtener una lista de elementos" o "iniciar sesión".
- **repository:** Esta es la interfaz que define los métodos que el useCase necesita para interactuar con los datos. Esta interfaz es implementada por repositoryImpl en la capa de datos.

- **CAPA “UI”**

Esta es la capa de presentación, que maneja la interfaz de usuario.

Contiene:

- **views:** Los componentes de la interfaz de usuario, como actividades, fragmentos, y vistas personalizadas. Esta capa interactúa con los casos de uso (useCase) para obtener datos y presentarlos al usuario.

7.5. Despliegue

Para el despliegue hemos decidido utilizar Azure debido a su sólida infraestructura en la nube, que ofrece escalabilidad, seguridad y una amplia gama de servicios integrados que pueden satisfacer las necesidades presentes y futuras de nuestro proyecto.



Azure proporciona soluciones flexibles y eficientes para el almacenamiento de datos y desarrollo de aplicaciones, permitiéndonos innovar rápidamente y optimizar nuestros procesos operativos. Además, la capacidad de Azure para integrarse con nuestras herramientas ([VSCode](#)), junto con su compromiso con la protección de datos de Europa, asegura que podamos mantener altos estándares de calidad y seguridad en el proyecto.

Hemos optado por implementar una arquitectura de microservicios en Azure. Utilizando servicios como Azure Container Registry, Azure App Web o Azure Blob Storage, podemos desarrollar, desplegar y gestionar componentes independientes que se comunican entre sí de manera eficiente. Esta estrategia nos permite realizar actualizaciones y escalar servicios específicos sin afectar el funcionamiento general del sistema, asegurando un desarrollo más ágil. A continuación, se muestra un esquema de comunicaciones entre servicios, y una explicación de cada servicio.

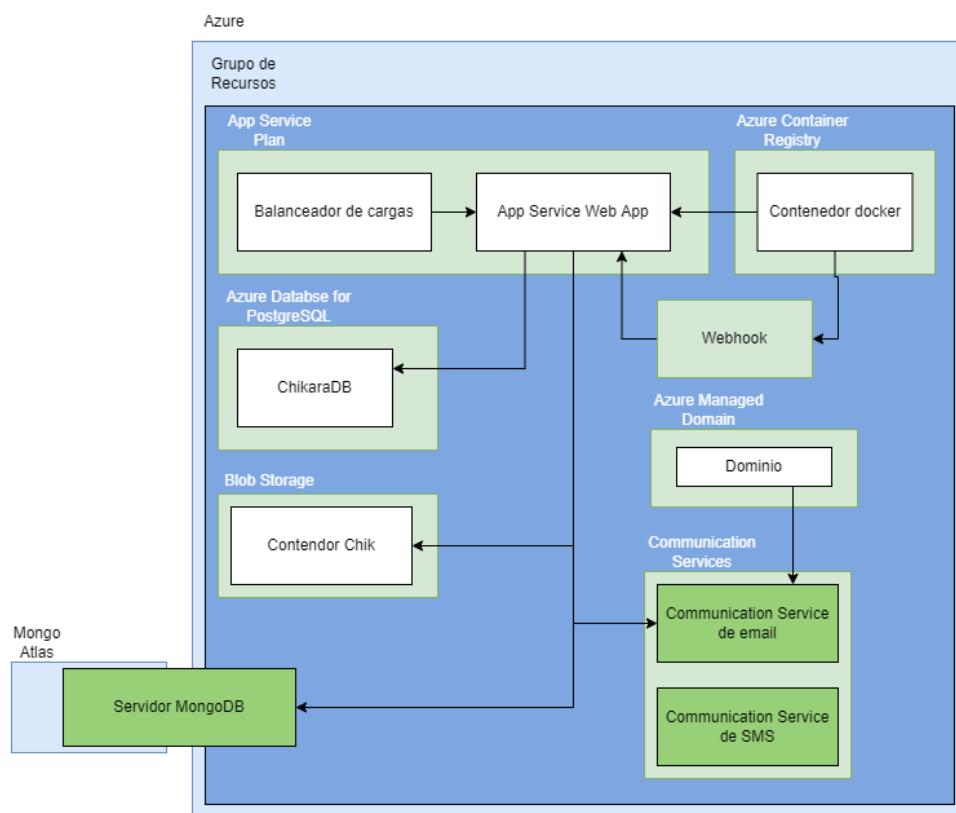


Diagrama 8 - Estructura Azure

| |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| Título: Proyecto Chikara |

- **Grupo de Recursos (Resource Group):** Es una agrupación lógica de los recursos de Azure utilizados en el proyecto.
- **App Service Plan:**
 - **Load Balancer:** Distribuye el tráfico de entrada entre varias instancias de la aplicación web para garantizar alta disponibilidad y escalabilidad.
 - **App Service Web App:** Aloja y ejecuta la aplicación web.
- **Azure Container Registry:**
 - **Docker Container:** Repositorio para almacenar y gestionar imágenes de contenedores Docker.
 - **Webhook:** Mecanismo para notificar eventos o actualizaciones desde el contenedor Docker hacia la aplicación web.
- **Azure Database for PostgreSQL:**
 - **ChikaraDB:** Base de datos PostgreSQL que almacena datos relacionales utilizados por la aplicación.
- **Blob Storage:**
 - **Chik Container:** Almacena objetos binarios grandes como imágenes, videos, etc.
- **Azure Managed Domain:**
 - **Dominio:** Servicio gestionado que proporciona dominio personalizado para la aplicación.
- **Communication Services:**
 - **Communication Service de email:** Servicio para enviar correos electrónicos.
 - **Communication Service de SMS:** Servicio para enviar mensajes de texto SMS.
- **Mongo Atlas:**
 - **Servidor MongoDB:** Base de datos NoSQL en MongoDB Atlas utilizada para el almacenamiento de datos no relacionales y que utiliza los servidores de Azure para funcionar.



7.5.1. CI/CD

Para un desarrollo y despliegue más ágiles, hemos implementado un sistema de CI/CD (integración y despliegue continuos por sus siglas en inglés) a través de los workers de GitLab. A continuación, se muestran los pasos a seguir.

- Incluir archivo “.gitlab.yml” al proyecto

Para empezar con la integración hay que programar los pasos que tiene que realizar GitLab, para ello, hay que configurar un archivo “.gitlab.yml” como el siguiente y hacer commit.

```
● ● ●
1 image: docker:latest
2 stages:
3 - build
4
5 variables:
6 DOCKER_IMAGE_NAME: chikara
7
8 services:
9 - docker:dind
10
11 build:
12 stage: build
13 script:
14 - apk add --no-cache bash curl
15 - curl --silent "https://gitlab.com/gitlab-org/incubation-engineering/mobile-devops/download-secure-files/-/raw/main/installer" | bash
16 - ls -lah .secure_files
17 - cp ./._secure_files/.env .env
18 - docker build -t $DOCKER_IMAGE_NAME .
19 - docker login $AZURE_REGISTRY -u $AZURE_REGISTRY_USERNAME -p $AZURE_REGISTRY_PASSWORD
20 - docker tag $DOCKER_IMAGE_NAME:latest $AZURE_REGISTRY/$DOCKER_IMAGE_NAME:latest
21 - docker push $AZURE_REGISTRY/$DOCKER_IMAGE_NAME:latest
22 only:
23 - main
24
```

Código 2 – Archivo .gitlab.yml

- Configurar GitLab

Aunque el archivo anterior indica a GitLab lo que debe hacer, hay que realizar dos configuraciones adicionales para que GitLab sea capaz de ejecutar la “pipeline” correctamente.

La primera es cargar las variables de entorno que necesarias para la ejecución, en nuestro caso son las siguientes.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Variables

[Collapse](#)

Variables store information that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more](#).

Variables can be accidentally exposed in a job log, or maliciously sent to a third party server. The masked variable feature can help reduce the risk of accidentally exposing variable values, but is not a guaranteed method to prevent malicious users from accessing variables. [How can I make my variables more secure?](#)

Variables can have several attributes. [Learn more](#).

- **Protected:** Only exposed to protected branches or protected tags.
- **Masked:** Hidden in job logs. Must match masking requirements.
- **Expanded:** Variables with \$ will be treated as the start of a reference to another variable.

| CI/CD Variables </> 3 | | Reveal values | Add variable |
|------------------------------------------------------------|-------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Key ↑ | Value | Environments | Actions |
| AZURE_REGISTRY | ***** | All (default) |   |
| AZURE_REGISTRY <small>Protected Masked</small> | | | |
| AZURE_REGISTRY_PASSWORD | ***** | All (default) |   |
| AZURE_REGISTRY_PASSWORD <small>Protected Masked</small> | | | |
| AZURE_REGISTRY_USERNAME | ***** | All (default) |   |
| AZURE_REGISTRY_USERNAME <small>Protected Masked</small> | | | |

Captura 2 - Variables de Entorno GitLab

El siguiente paso es incluir en los secure files nuestro archivo de variables de entorno de FastAPI, para que Gitlab incluya en la imagen de Docker que despliega en Azure dicho archivo.

Secure Files

[Collapse](#)

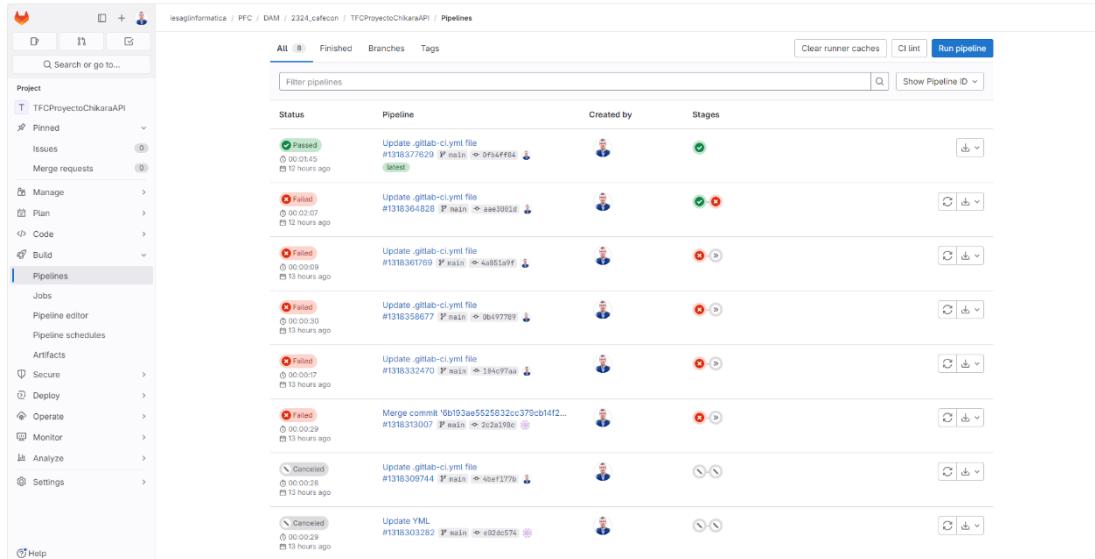
Use Secure Files to store files used by your pipelines such as Android keystores, or Apple provisioning profiles and signing certificates. [Learn more](#)

| Files | 1 | Upload File |
|-----------|---------------|---------------------------------------------------------------------------------------|
| File name | Uploaded date | Actions |
| .env | 13 hours ago |  |

Captura 1 – Secure Files Gitlab

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Siguiendo los pasos anteriores Gitlab es capaz de realizar un despliegue automático en Azure cada vez que detecta un commit nuevo en la rama main.



| Status | Pipeline | Created by | Stages |
|---------------------------------------------|---------------------------------------------------------|------------|------------------------------------------------------|
| Passed | Update_gitlab-ci.yml file #1318377629 | 9fb44f94 | [green checkmark] |
| Failed | Update_gitlab-ci.yml file #1318364828 | aee30916 | [red circle] |
| Failed | Update_gitlab-ci.yml file #1318350769 | 4a851a9f | [red circle] |
| Failed | Update_gitlab-ci.yml file #1318358677 | 0b497789 | [red circle] |
| Failed | Update_gitlab-ci.yml file #1318332470 | 184c97aa | [red circle] |
| Failed | Merge commit '0b103ae5525932cc370cb14f2...' #1318313007 | 2c261986 | [red circle] |
| Cancelled | Update_YML #1318309744 | 40e1377b | [cancel icon] |
| Cancelled | Update_YML #1318303282 | e02de574 | [cancel icon] |

Captura 3 - Pipelines GitLab

IMPLEMENTACIÓN

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

8. IMPLEMENTACIÓN

8.1. Frontend

CAPA DATA:

Esta capa es responsable de manejar los datos de la aplicación, ya sea desde una base de datos local, una API remota, o cualquier otra fuente de datos.

NetworkModule:

Clase que configura y proporciona los componentes necesarios para la conexión a una API usando Retrofit. Su implementación se realiza de la siguiente manera (ver [Código 3 - NetworkModule](#)):

La clase está anotada con `@Module` y `@InstallIn(SingletonComponent::class)`, lo que indica que es un módulo de Dagger Hilt y que sus dependencias estarán disponibles en un componente Singleton, es decir, una única instancia para toda la aplicación.

AuthKey: variable que se usa para almacenar una clave de autenticación que se incluirá en los encabezados de las solicitudes, utilizando un JWT ([9.1 Securización](#)).

Todos los métodos dentro de esta clase tienen la notación `@Provides`. En Dagger Hilt se utiliza para indicar que un método dentro de un módulo proporciona una instancia de un tipo de dependencia. Es decir, este método es responsable de crear y configurar esa instancia cuando sea necesario en la aplicación.

Método providesRetrofit: Este método crea y configura una instancia de Retrofit.

- Utiliza `BASE_URL` como la URL base para las solicitudes, que se obtiene de la configuración de Gradle.
- Configura un `OkHttpClient` con un interceptor que añade encabezados (incluyendo la clave de autenticación, ID, modelo y marca del teléfono) a cada solicitud.



Método `providesOkHttpClient`: Crea un `OkHttpClient` con un interceptor de logging (`HttpLoggingInterceptor`) que captura y registra los detalles de las respuestas de la red, lo que es útil para el debugging.

Esta clase (`NetworkModule`) provee (`@Provide`) todos los servicios de Api e implementación de repositorios para el funcionamiento de las consultas.

```
1 /**
2  * Tiene la conexion a la API utilizando retrofit
3  * Tipo de objeto SINGELTON :D
4 */
5 @Module
6 @InstallIn(SingletonComponent::class)
7 object NetworkModule {
8
9     // Auth Key Para poder acceder a los endpoints de la App
10    var AuthKey: String = ""
11
12    /**
13     * SINGELTON
14     * Crea la conexion necesaria a la API
15     * Recibe como parametro un interceptor para obtener la informacion de la consulta.
16     *
17     * BASE_URL VIENE DEL GRADLE, POR EL TIPO DE CONFIGURACION!
18     */
19    @Provides
20    @Singleton
21    fun providesRetrofit(okHttpClient: OkHttpClient): Retrofit {
22
23        return Retrofit.Builder()
24            .baseUrl(BASE_URL)
25            .addConverterFactory(GsonConverterFactory.create())
26            .client(okHttpClient)
27            .client(OkHttpClient.Builder().addInterceptor { chain ->
28                val request = chain.request().newBuilder()
29                .addHeader("authorization", "Bearer ${AuthKey}")
30                .addHeader("phone_id", android.os.Build.ID)
31                .addHeader("phone_model", android.os.Build.MODEL)
32                .addHeader("phone_brand", android.os.Build.BRAND)
33                .build()
34
35                chain.proceed(request)
36            }.build())
37            .build()
38    }
39
40    /**
41     * Interceptor se utilizan para obtener toda la respuesta de la Red,
42     * La guarda en el LOG permite ver que pasa con las llamadas
43     * Segun el nivel podemos obtener mas o menos info
44     */
45    @Provides
46    @Singleton
47    fun providesOkHttpClient(): OkHttpClient {
48        //Este nivel de interceptor captura toda la respuesta de la Api
49        val interceptor = HttpLoggingInterceptor().setLevel(HttpLoggingInterceptor.Level.BODY)
50
51        return OkHttpClient
52            .Builder()
53            .addInterceptor(interceptor)
54            .build()
55    }
56
57    /**
58     * Construye el repositorio de API SERVICE
59     */
60    @Provides
61    fun providerLogin ApiService(retrofit: Retrofit): Login ApiService {
62        return retrofit.create(Login ApiService::class.java)
63    }
64
65    /**
66     * Para diferenciar entre repositorios utilizamos la Notacion @Name
67     */
68    //@Named
69    @Provides
70    fun provideLoginRepository(apiService: Login ApiService): Login Repository {
71        return Login RepositoryImpl(apiService)
72    }
}
```

Código 3 - `NetworkModule`



Package Service:

Agrupan los servicios que interactúan con la API, Ejemplo:

Register ApiService:

Interfaz encargada de definir los métodos para interactuar con la API, encargándose de las peticiones HTTP del registro. Estos métodos son funciones suspendidas de Kotlin, lo que permite su uso en coroutines para realizar llamadas de red de manera asíncrona (ver Código 4 - Register ApiService)

Método `userNameExist`: Que realiza una solicitud GET al endpoint “/users/searchuser” con un parámetro de consulta user para comprobar si un nombre de usuario ya existe.

Método `emailExist`: Que realiza una solicitud GET al endpoint “/users/searchemail” con un parámetro de consulta email para comprobar si un correo electrónico ya existe.

Método `registerUser`: Que realiza una solicitud POST al endpoint “/users/register” incluyendo en el body de la request un `userDto` para registrar un nuevo usuario.

```
1 interface Register ApiService {  
2  
3     @GET("/users/searchuser")  
4     suspend fun userNameExist(@Query("user") userName:String):Boolean  
5  
6     @GET("/users/searchemail")  
7     suspend fun emailExist(@Query("email") email:String):Boolean  
8  
9     @POST("/users/register")  
10    suspend fun registerUser(@Body userDto:UserDto):Boolean  
11  
12 }
```

Código 4 - Register ApiService

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Package RepositoryImpl:

Paquete que agrupa las implementaciones de los repositorios a modo de ejemplo podemos ver la siguiente implementación.

RegisterRepositoryImpl:

La clase es una implementación del repositorio RegisterRepository que interactúa con un servicio de API para registrar usuarios y verificar la existencia de nombres de usuario y correos electrónicos. Aquí está el resumen de su funcionamiento (ver [Código 5 - RegisterRepositoryImpl](#)):

Inyección de Dependencias: Utiliza la anotación `@Inject` para inyectar una instancia de `Register ApiService`, que proporciona los métodos para hacer las llamadas a la API.

Método `userNameExist`: Llama al método correspondiente del `Register ApiService` para verificar si un nombre de usuario ya existe. Si la llamada tiene éxito, retorna true; si falla, retorna false.

Método `emailExist`: Llama al método correspondiente del `Register ApiService` para verificar si un correo electrónico ya existe. Si la llamada tiene éxito, retorna true; si falla, retorna false.

Método `registerUser`: Llama al método correspondiente del `Register ApiService` para registrar un nuevo usuario. Si la llamada tiene éxito, retorna true; si falla, retorna false.



```
1 class RegisterRepositoryImpl @Inject constructor(private val apiService: Register ApiService): RegisterRepository {
2
3     /**
4      * Comprueba que el UserName existe
5      * True si existe
6      */
7     override suspend fun userNameExist(userName: String): Boolean {
8         runCatching {
9             apiService.userNameExist(userName)
10        }
11        .onSuccess {
12            Log.i("RegistroUsuario: ", "API Usuario Existe = $it")
13            return true
14        }
15        . onFailure {
16            Log.i("RegistroUsuario: ", "API Usuario Existe = $it")
17        }
18    }
19    return false
20 }
21 /**
22 * Comprueba que el Email existe
23 * True si existe
24 */
25 override suspend fun emailExist(email: String): Boolean {
26     runCatching {
27         apiService.emailExist(email)
28     }
29     .onSuccess {
30         return true
31     }
32     . onFailure { }
33
34     return false
35 }
36 /**
37 * Intenta registrar un Nuevo Usuario.
38 * Devuelve el DTO
39 */
40 override suspend fun registerUser(userDto: UserDto): Boolean {
41     runCatching {
42         apiService.registerUser(userDto)
43     }
44     .onSuccess {
45         Log.i("RegistroUsuario: ", "Registro de usuario satisfactorio $it")
46         return true
47     }
48     . onFailure {
49         Log.i("RegistroUsuario: ", "Error registro de usuario " + it.message)
50     }
51 }
52 return false
53 }
54 }
55 }
56 }
57 }
```

Código 5 - RegisterRepositoryImpl

CAPA DOMAIN:

Esta capa contiene la lógica de negocio de la aplicación y actúa como un intermediario entre la capa de datos y la capa de UI. Incluye:

Package Model:

Este paquete contiene elementos “**data class**”, que en Kotlin se utiliza para representar objetos con propiedades. En este caso utilizamos DTO ([Data Transfer Object](#), por sus siglas en inglés) es un objeto que se utiliza para transportar datos



entre diferentes partes de un sistema, especialmente entre las capas de una aplicación o entre aplicaciones distintas. Los DTOs se usan para simplificar la transferencia de datos y minimizar el número de llamadas entre sistemas. Como ejemplo tenemos:

ChickDto:

Esta clase de datos que agrupa las propiedades relacionadas con un Chick, incluyendo información básica como el título y autor, metadatos como la fecha y número de likes, y colecciones de contenido, comentarios y menciones asociadas.

```
1  data class ChickDto (
2
3      var _id:String = "",
4      val title : String,
5      val author : String? = null,
6      val date : String? = null,
7      var likes : Long = 0,
8      val isprivate : Boolean,
9      var content : List<ChickContentDto> = emptyList(),
10     var comments : List<ChickCommentDto> = emptyList(),
11     var mencias : List<String> = emptyList()
12
13 )
```

Código 6 - ChickDto

Package Repository:

Paquete en el que se definen clases y sus métodos, los cuales son implementados por otras clases.

[ChickRepository](#): define los métodos para interactuar con datos relacionados con Chicks. Aquí está un resumen de su funcionamiento general:



Método getTopChicks: Este método busca y devuelve una lista de ChickDto. Si la operación tiene éxito, devuelve la lista (que puede estar vacía si no hay Chicks). En caso de error, devuelve una lista vacía.

Método findByAuthor: Este método busca y devuelve una lista de ChickDTO que pertenecen a un autor específico identificado por su userId. Si la operación tiene éxito, devuelve la lista (que puede estar vacía). En caso de error, devuelve una lista vacía.

Método newChick: Este método añade un nuevo Chick a la colección. Recibe un objeto ChickDTO que representa los datos del nuevo Chick y devuelve un valor booleano indicando si la operación fue exitosa (true) o no (false).

```
1 interface ChickRepository {
2
3     /**
4      * Busca el TOP de chicks por numero de likes
5      * si lo consigue devuelve la lista. (puede estar vacia)
6      * en caso de error devuelve una lista vacia.
7      */
8     suspend fun getTopChicks():List<ChickDto>
9
10    /**
11     * Busca los Chicks por el Id del Author
12     * si lo consigue devuelve la lista. (puede estar vacia)
13     * en caso de error devuelve una lista vacia.
14     */
15    suspend fun findByAuthor(userId:String):List<ChickDto>
16
17    /**
18     * Add new Chick
19     */
20    suspend fun newChick(chickDto: ChickDto):Boolean
21
22 }
```

Código 7 - ChickRepository

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Package UseCase:

Este paquete contiene los casos de uso para la relación con los repositorios al realizar las operaciones de consulta. El siguiente usecase explica el su funcionamiento:

ChickUseCases:

Utiliza la anotación `@Inject` para inyectar una instancia de ChickRepository, facilitando la gestión de dependencias (ver [Código 8 - ChickUseCases](#)).

Método `getTopChicks`: Llama al método correspondiente del repositorio (`getTopChicks`) para obtener una lista de ChickDTO. Devuelve esta lista, que puede estar vacía si no se encuentran Chicks o si ocurre un error.

Método `findByAuthor`: Llama al método correspondiente del repositorio (`findByAuthor`) para obtener una lista de ChickDTO que pertenecen a un autor específico, identificado por `userId`. Devuelve esta lista, que puede estar vacía si no se encuentran Chicks o si ocurre un error.

Método `newChick`: Este método realiza varias operaciones antes de llamar al repositorio para añadir un nuevo Chick.

Recorre los elementos de contenido del ChickDTO y, si un elemento es una imagen (`ChickTypeDto.TYPE_IMG`), convierte su valor a Base64.

Finalmente, llama al método `newChick` del repositorio para añadir el nuevo Chick y devuelve un valor booleano indicando si la operación fue exitosa (true) o no (false).



```
1  class ChickUseCases @Inject constructor(private val repository: ChickRepository) {
2
3      /**
4       * Busca el TOP de chicks por numero de likes
5       * si lo consigue devuelve la lista. (puede estar vacia)
6       * en caso de error devuelve una lista vacia.
7       */
8      suspend fun getTopChicks() = repository.getTopChicks()
9
10     /**
11      * Busca los Chicks por el Id del Author
12      * si lo consigue devuelve la lista. (puede estar vacia)
13      * en caso de error devuelve una lista vacia.
14      */
15     suspend fun findByAuthor(userId: String) = repository.findByAuthor(userId)
16
17     /**
18      * Add new Chick
19      */
20     suspend fun newChick(context: Context, chickDto: ChickDto): Boolean {
21
22         val encodeBase64 = EncodeBase64()
23
24         //For Each Element in the Dto check if is Image and Encode to Base64
25         chickDto.content.forEach { cont ->
26             if (cont.type == ChickTypeDto.TYPE_IMG) {
27                 val uri: Uri = Uri.parse(cont.value)
28                 cont.value = encodeBase64(context, uri)
29             }
30         }
31
32         return repository.newChick(chickDto)
33     }
34 }
35 }
```

Código 8 - ChickUseCases

CAPA UI

Esta es la capa de presentación, que maneja la interfaz de usuario. Contiene paquetes que representarán las vistas y todo lo necesario para su funcionamiento. Como ejemplo revisaremos el siguiente paquete:

Package Welcome:

Contiene la clase `WelcomeActivity` que se ocupa de controlar la vista, `WelcomeState` que representa los estados que puede tener la Activity y `WelcomeViewModel` que se ocupa de comunicar a la Activity el estado actual para que ésta realice acciones.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

WelcomeActivity:

Esta clase utiliza la arquitectura de Jetpack y Dagger Hilt para la inyección de dependencias y gestión de estados. Este es el resumen de su funcionamiento general:

Anotación @AndroidEntryPoint: Indica que Dagger Hilt debe inyectar dependencias en esta actividad.

binding: Se usa para acceder a los elementos definidos en el layout (XML que se utiliza para crear la interfaz gráfica y definir los elementos).

welcomeViewModel: Instancia de WelcomeViewModel, que se usa para gestionar el estado de la UI.

Método onCreate: Este método se llama cuando se crea la actividad. Inicializa el binding según el layout y establece la vista principal. Por último, se llama al método initUI para inicializar la interfaz de usuario.

Método initUI: Inicializa todos los procesos necesarios para que la vista funcione, llamando a initUIlaunchLoginFlow e initUIState.

Método initUIlaunchLoginFlow: Llama a launchLoginFlow del welcomeViewModel para iniciar el proceso de inicio de sesión utilizando el contexto de la aplicación.

Método initUIState: Utiliza lifecycleScope.launch para iniciar una coroutine que observa el ciclo de vida de la actividad.

- Usa repeatOnLifecycle para escuchar los cambios de estado de la actividad.
- Recoge los estados del welcomeViewModel.state y llama a métodos específicos dependiendo del estado Loading, Error, Success según estos estados se llaman a sus correspondientes métodos:



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

- **successState**: Se llama cuando el estado es Success. Actualiza la UI para mostrar un mensaje de bienvenida y, después de un retraso de 2 segundos, navega a HomeActivity.
- **errorState**: Se llama cuando el estado es Error. Actualiza la UI para mostrar un mensaje de error y, después de un retraso de 2 segundos, navega a LoginActivity.
- **loadingState**: Se llama cuando el estado es Loading. No realiza ninguna acción específica en este caso.



```
1  @AndroidEntryPoint
2  class WelcomeActivity : AppCompatActivity() {
3
4      private lateinit var binding: ActivityWelcomeBinding
5
6      //Necesario para Que initUIstate Funcione
7      private val welcomeViewModel:WelcomeViewModel by viewModels()
8      override fun onCreate(savedInstanceState: Bundle?) {
9          super.onCreate(savedInstanceState)
10
11         binding = ActivityWelcomeBinding.inflate(layoutInflater)
12         setContentView(binding.root)
13
14         initUI()
15     }
16
17     /**
18      * Inicializa todos los procesos necesarios para que la vista funcione.
19      */
20     private fun initUI() {
21         initUIState()
22         initUIlaunchLoginFlow()
23     }
24
25     /**
26      * Lee desde el fichero de properties e intenta logear.
27      */
28     private fun initUIlaunchLoginFlow() {
29         welcomeViewModel.launchLoginFlow(applicationContext)
30     }
31
32     /**
33      * Inicializa el controlador para el cambio de estados.
34      */
35     private fun initUIState() {
36         //Hilo que esta pendiente de la vida de la VIEW, si la view muere el para!
37         lifecycleScope.launch {
38             repeatOnLifecycle(Lifecycle.State.STARTED) {
39                 welcomeViewModel.state.collect {
40                     //Siempre que cambien el estado hara lo siguiente
41                     when (it) {
42                         //It es la informacion del estado que puede contener informacion.
43                         //Cada cambio de estado llama a su metodo
44                         WelcomeState.Loading -> loadingState()
45                         is WelcomeState.Error -> errorState(it)
46                         is WelcomeState.Success -> successState(it)
47                     }
48                 }
49             }
50         }
51     }
52
53     /**
54      * Cambio de Estado a Success.
55      */
56     private suspend fun successState(it: WelcomeState.Success) {
57         binding.pbWelcome.isVisible = false
58         binding.tvWelcomeMessage.text = getText(R.string.text_welcome_message2)
59
60         delay(2000L)
61         val intent = Intent(this, HomeActivity::class.java)
62         startActivity(intent)
63     }
64
65     /**
66      * Cambio de Estado a Error.
67      * En caso de error te envia al Loggin, espera unos segundos
68      */
69     private suspend fun errorState(it: WelcomeState.Error) {
70         binding.pbWelcome.isVisible = false
71         binding.tvWelcomeMessage.text = getText(R.string.text_welcome_message)
72
73         delay(2000L)
74         val intent = Intent(this, LoginActivity::class.java)
75         startActivity(intent)
76     }
77
78     /**
79      * Cambio de Estado a Loading.
80      * No tiene que hacer nada en principio.
81      */
82     private fun loadingState() {
83     }
84 }
```

Código 9 - WelcomeActivity

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

WelcomeState:

Es una clase sellada ([sealed class](#)) que se utiliza para representar diferentes estados en la lógica de la aplicación. En esta se definen un número limitado de subclases. Permite gestionar los estados con un control exhaustivo utilizando la expresión “when”. Sus estados definidos son los siguientes:

[Loading](#): Representa el estado de carga. Es un Data Object, lo que significa que es un objeto singleton sin propiedades. Se usa cuando la aplicación está en proceso de cargar datos o realizar alguna operación.

[Error](#): Representa el estado de error. Es un data class que contiene una propiedad error de tipo Boolean. Se utiliza para informar que un error ha ocurrido.

[Success](#): Representa el estado de éxito. Es un data class que contiene una propiedad success de tipo Boolean. Se utiliza para informar sobre el éxito de la operación.



```

1  sealed class WelcomeState {
2
3      //Se ocupa del Estado de Loading
4      data object Loading: WelcomeState()
5
6      //Al cambiar el estado a Error enviara los datos que sean necesarios
7      data class Error(val error:Boolean): WelcomeState()
8
9      //Al cambiar al estado de Success enviara los datos que necesitemos.
10     data class Success(val success:Boolean): WelcomeState()
11 }

```

Código 10 - WelcomeState

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

WelcomeViewModel:

Es una implementación de ViewModel que se utiliza para gestionar la lógica y el estado de la pantalla de Welcome de la aplicación, utilizando Dagger Hilt para la inyección de dependencias y coroutines para la ejecución de tareas en segundo plano. Funcionamiento (ver [Código 11 - WelcomeViewModel](#)):

Anotación @HiltViewModel: Indica que Hilt debe manejar la creación y la inyección de dependencias en esta clase ViewModel.

Inyección GetLoginUseCase: Utiliza la anotación `@Inject` en el constructor para inyectar una instancia de GetLoginUseCase, que se utiliza para realizar la operación de inicio de sesión.

Estado Mutable: `_state: Un MutableStateFlow que mantiene el estado actual de la vista (WelcomeState), con un valor inicial de Loading.`

state: Un StateFlow público y de solo lectura que expone el estado actual de la vista a otras clases.

userPreferences: Instancia de UserPreferences, utilizada para acceder a las configuraciones del usuario.

Método launchLoginFlow: Inicializa userPreferences con el contexto de la aplicación. Lanza una coroutine en viewModelScope para ejecutar el flujo de inicio de sesión. Actualiza el estado a Loading. Llama a tryToLoggin para intentar iniciar sesión y actualiza el estado a Success o Error según el resultado.

Método tryToLoggin: Se ejecuta en un contexto de IO (hilo de entrada/salida) para operaciones de larga duración. Obtiene las configuraciones de usuario de userPreferences. Verifica que las configuraciones no sean nulas y que el nombre de usuario y la contraseña no estén en blanco.

Llama a getLoginUseCase con el nombre de usuario y la contraseña para intentar iniciar sesión, devolviendo true si tiene éxito y false en caso contrario. Para actualizar el estado.



```
1  @HiltViewModel
2  class WelcomeViewModel @Inject constructor(private val getLoginUseCase: GetLoginUseCase) : ViewModel(){
3
4      //Como valor inicial le paso el estado de Cargando
5      private var _state = MutableStateFlow<WelcomeState>(WelcomeState.Loading)
6      //devuelve el estado en el que se encuentra, por eso no es publica
7      val state: StateFlow<WelcomeState> = _state
8
9      lateinit var userPreferences:UserPreferences
10
11     /**
12      * Inicializa el Flow para la ventana de carga y lanza el Loggin esperando respuesta.
13      */
14     fun launchLoginFlow(context: Context){
15         userPreferences = UserPreferences(context)
16         //Esta trabajando en el Hilo principal
17         viewModelScope.launch {
18
19             //Cambio el stado a loading.
20             _state.value = WelcomeState.Loading
21
22             //Esperando la respuesta
23             val result = tryToLogin()
24
25             //si la respuesta es correcta
26             if (result){
27                 //Paso el esdado a Success y le paso el texto del horoscopo
28                 //SE PUEDE DEVOLVER EL OBJETO ENTERO O LO QUE NECESITEMOS.
29                 _state.value = WelcomeState.Success(true)
30             } else{
31                 _state.value = WelcomeState.Error(false)
32             }
33         }
34     }
35
36     /**
37      * Intenta logear obteniendo los datos de userPreferences.
38      * devuelve True si lo Logra.
39      * Si no tiene datos o no lo consigue devuelve false
40      */
41     suspend fun tryToLogin(): Boolean {
42         return withContext(Dispatchers.IO) {
43             val userPreferenceModel = userPreferences.getSettings().first()
44
45             //Comprueba que los UserPreference no es null para leer los datos
46             if (userPreferenceModel != null) {
47                 val user = userPreferenceModel.userName
48                 val password = userPreferenceModel.password
49
50                 //Comprueba que los datos obtenidos no son falsos.
51                 if (user.isNotBlank() && password.isNotBlank()){
52                     return@withContext getLoginUseCase(user, password)
53                 }else{
54                     return@withContext false
55                 }
56
57             } else {
58                 return@withContext false
59             }
60         }
61     }
62 }
63 }
```

Código 11 - WelcomeViewModel

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Package Utils:

Fuera de las capas tenemos este paquete, que contiene clases y funciones utilitarias genéricas que pueden ser utilizadas por cualquier parte de la aplicación. Esto ayuda a mantener el código limpio, organizado y libre de redundancias. A modo de ejemplo tenemos:

UserPreferences:

Clase encargada de manejar las preferencias del usuario utilizando Jetpack DataStore, para almacenar y recuperar los datos de configuración. Su funcionamiento es el siguiente (ver [Código 12 - UserPreferences](#)):

LAYOUT_PREFERENCES_NAME: Nombre del archivo de preferencias, en este caso "[ChikaraSettings](#)".

Extensión dataStore: Una propiedad de extensión para el Context que define un DataStore de tipo Preferences, utilizando el nombre especificado en LAYOUT_PREFERENCES_NAME.

Class UserPreferences: Utiliza la anotación [@Inject](#) para permitir la inyección de dependencias de Dagger Hilt, obteniendo el Context necesario para acceder al DataStore.

KEY_USER_STR y **KEY_PASSWORD_STR**: Claves de preferencias definidas usando stringPreferencesKey para almacenar y recuperar el nombre de usuario y la contraseña.

savePreference: Método genérico para guardar una preferencia. Toma una clave de preferencia (preferenceKey) y un valor (value), y los guarda en el DataStore.

deletePreference: Método genérico para eliminar una preferencia. Toma una clave de preferencia (preferenceKey) y la elimina del DataStore.



`getSettings`: Método para leer las preferencias del usuario y devolverlas como un flujo (Flow) de UserPreferencesModel. Si las preferencias no existen, se devuelven valores por defecto.

```
1 //Nombre de los settings
2 private const val LAYOUT_PREFERENCES_NAME = "ChikaraSettings"
3
4 //Data Store para los settings
5 val Context.dataStore: DataStore<Preferences> by preferencesDataStore(name = LAYOUT_PREFERENCES_NAME)
6
7 class UserPreferences @Inject constructor(private val context: Context) {
8
9     /**
10      * Listado de KEY name and Type of User Preferences.
11      */
12     companion object {
13         val KEY_USER_STR = stringPreferencesKey("User")
14         val KEY_PASSWORD_STR = stringPreferencesKey("Password")
15     }
16
17     /**
18      * Metodo generico para añadir
19      */
20     suspend fun <T> savePreference(preferenceKey: Preferences.Key<T>, value:I) {
21         context.dataStore.edit { preference ->
22             preference[preferenceKey] = value
23         }
24     }
25
26     /**
27      * Metodo Generico para borrar
28      */
29     suspend fun <T> deletePreference(preferenceKey: Preferences.Key<T>) {
30         context.dataStore.edit { preference ->
31             preference.remove(preferenceKey)
32         }
33     }
34
35     /**
36      * Lee todos los datos y los devuelve,
37      * Si no tiene datos por defecto pone ( ?: "" ) vacio,
38      */
39     public fun getSettings(): Flow<UserPreferencesModel> {
40         return context.dataStore.data.map { preferences ->
41             UserPreferencesModel(
42                 userName = preferences[KEY_USER_STR] ?: "",
43                 password = preferences[KEY_PASSWORD_STR] ?: ""
44             )
45         }
46     }
47 }
48 }
```

Código 12 - `UserPreferences`



8.2. Backend

El backend está construido con FastAPI, es un moderno framework web para Python, diseñado para construir APIs rápidas y de alto rendimiento con facilidad. Aprovechando las capacidades de tipo de Python, FastAPI proporciona una validación automática de datos, generación de documentación interactiva y rendimiento optimizado, comparable al de los frameworks más rápidos disponibles.

Entender los códigos más importantes de FastAPI es esencial para aprovechar al máximo sus beneficios. Desde la creación de endpoints que gestionan las solicitudes y respuestas http, hasta los modelos ORM, estos códigos forman la base de una aplicación robusta y eficiente. A continuación, revisaremos algunos de los elementos más importantes y utilizados.

ENDPOINTS

Los Endpoints son fundamentales para la comunicación entre el cliente y el servidor en una arquitectura API Rest. Permiten la modularidad, facilitando la interacción entre diferentes sistemas y servicios.

Cada endpoint representa una URL específica que responde a solicitudes HTTP (como **GET, POST, PUT, DELETE**) y se asocia con una funcionalidad particular del servicio.

El siguiente código corresponde a un endpoint que permite el inicio de sesión en Chikara y devuelve JWT si las credenciales son correctas.

```
1 from services.user_dataService import User_DataService
2 from config.db_depend import get_db
3 from config.jwt import generate_token
4
5
6 router= APIRouter()
7
8 @cbv(router)
9 class user_dataController:
10
11     def __init__(self):
12         self.service = User_DataService()
13
14     @router.get("/login",response_model=str,status_code=200)
15     def UserLogin(self,request:Request,user: str, password: str, db: Session = Depends(get_db)):
16         result=self.service.getUserLogin(db,request, user, password)
17         if result is None:
18             raise HTTPException(status_code=404,detail="Not Found")
19         return generate_token(result.dict())
```

Código 13 - Endpoint



SERVICES

Un Service es un componente que encapsula la lógica de negocio y operaciones específicas que la aplicación necesita realizar. La idea detrás de un servicio es separar esta lógica de otras capas de la aplicación, como la capa de presentación ([controladores](#)) o la capa de datos ([repositorios](#)), promoviendo una arquitectura más modular, mantenible y escalable.

El siguiente código representa la lógica del inicio de sesión en chikara.

```
● ● ●
1  from config.email import sendEmailBlock, sendEmailRecovery, sendEmailUnBlock, sendEmailWelcome
2  from models.user_data import user_data
3  from repository.user_dataRepository import User_DataRepository
4
5
6  class User_DataService():
7
8      def __init__(self):
9          self.repository = User_DataRepository()
10
11     def getAllUsers(self, db: Session):
12         return self.repository.get_all(db)
13
14     def getUserLogin(self, db: Session, request: Request, user: str, password: str):
15         result,device,bloqueado=self.repository.get_user_login(db,request, user, password)
16         if bloqueado:
17             hiloemail = threading.Thread(target=sendEmailUnBlock, args=(result,device))
18             hiloemail.start()
19             return None
20         if device is not None:
21             #Enviamos email de inicio de sesion
22             hiloemail = threading.Thread(target=sendEmailBlock, args=(result,device))
23             hiloemail.start()
24
25     return result
```

Código 14 - Service

REPOSITORIES Y MODELS

Los repositories ([repositorios](#)) encapsulan la lógica para acceder a los datos. Proporcionan una capa de abstracción entre la lógica de negocio y la base de datos. Esto permite que la lógica de negocio interactúe con los datos sin necesidad de conocer los detalles específicos de cómo se almacenan y recuperan.



A su vez los modelos son una representación de una tabla de base de datos en forma de clase de Python, utilizada en el contexto del mapeo objeto-relacional (ORM). Cada atributo de la clase corresponde a una columna en la tabla de la base de datos, y las instancias de la clase representan filas individuales de la tabla.

Para definirlos hay múltiples dependencias, pero la que hemos usado nosotros es SQLAlchemy. En las siguientes capturas se puede ver el modelo user y el repositorio para hacer login.

```
● ● ●
1 from sqlalchemy import Column, Integer, String, Date, DateTime, Boolean, func
2 from sqlalchemy.dialects.postgresql import UUID
3 import uuid
4
5 from config.db import Base
6
7 class user_data(Base):
8     __tablename__ = 'user_data'
9
10    id = Column('id_user',UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
11    user_name = Column(String(150), nullable=False)
12    email = Column(String(300), nullable=False)
13    pwd = Column(String(300), nullable=False)
14    first_name = Column(String(150), nullable=False)
15    first_last_name = Column(String(150), nullable=False)
16    second_last_name = Column(String(150))
17    birthdate = Column(Date, nullable=False)
18    account_creation = Column(DateTime(timezone=True), default=func.now())
19    is_premium = Column(Boolean, default=False)
20
21    def __repr__(self):
22        return f"<User(id_user={self.id}, user_name='{self.user_name}', "
23                    f"email='{self.email}', password='{self.pwd}', "
24                    f"first_name='{self.first_name}', first_last_name='{self.first_last_name}', "
25                    f"second_last_name='{self.second_last_name}', birthdate='{self.birthdate}', "
26                    f"account_creation='{self.account_creation}', is_premium={self.is_premium})>"
27
28    def dict(self):
29        return {
30            "id_user": str(self.id),
31            "user_name": self.user_name,
32            "email": self.email,
33            "first_name": self.first_name,
34            "first_last_name": self.first_last_name,
35            "second_last_name": self.second_last_name,
36            "birthdate": str(self.birthdate),
37            "account_creation": str(self.account_creation),
38            "is_premium": self.is_premium
39        }
```

Código 15 - Modelo User



Ciclo: Desarrollo de Aplicaciones Multiplataforma

Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

```
1 from fastapi import Request
2 from models.user_data import user_data
3 from models.user_devices import user_devices
4 from models.user_log import user_log
5 from repository.AbstractRepository import AbstractRepository
6 from repository.user_logRepository import User_LogRepository
7 from repository.user_devicesRepository import user_devicesRepository
8
9
10 class User_DataRepository(AbstractRepository):
11     def __init__(self):
12         self.entity = user_data
13         self.user_logRepository = User_LogRepository()
14         self.user_devicesRepository = user_devicesRepository()
15
16     # Sin cifrar a md5 la password
17     def get_user_login(self, db, request: Request, user: str, password: str):
18         result = None
19         device = None
20
21         if user[0] == '@': #Comprueba si el email empieza por @, si es así, es un username
22             result = db.query(self.entity).filter(self.entity.user_name == user, self.entity.pwd == password).first()
23
24             #Crea un nuevo objeto user_log con el id del usuario
25             userlognew = user_log(id_user=result.id)
26             #Añade un nuevo registro en la tabla user_log
27             self.user_logRepository.add(userlognew, db)
28
29             #Comprueba si el phone_id no está en la tabla user_devices
30             device_db = db.query(user_devices).filter(user_devices.phone_id == request.headers.get("phone_id")).filter(user_devices.id_user == result.id).first()
31
32             if device_db is None:
33                 #Crea un nuevo objeto user_devices con el phone_id y el id del usuario
34                 userdevicesnew = user_devices(phone_id=request.headers.get("phone_id"), id_user=result.id,
35                                               phone_model=request.headers.get("phone_model"), phone_brand=request.headers.get("phone_brand"))
36                 #Añade un nuevo registro en la tabla user_devices
37                 device = self.user_devicesRepository.add(userdevicesnew, db)
38                 return result, device, True
39             else:
40                 #Comprueba si el dispositivo está bloqueado
41                 if device_db.block == True:
42                     return result, device_db, True
43                 return result, None, False
44             else:
45                 return None, None, False
46
47         else:
48             result = db.query(self.entity).filter(self.entity.email == user, self.entity.pwd == password).first()
49
50             if result:
51                 #Crea un nuevo objeto user_log con el id del usuario
52                 userlognew = user_log(id_user=result.id)
53                 #Añade un nuevo registro en la tabla user_log
54                 self.user_logRepository.add(userlognew, db)
55
56                 #Comprueba si el phone_id no está en la tabla user_devices
57                 device_db = db.query(user_devices).filter(user_devices.phone_id == request.headers.get("phone_id")).filter(user_devices.id_user == result.id).first()
58
59                 if device_db is None:
60                     #Crea un nuevo objeto user_devices con el phone_id y el id del usuario
61                     userdevicesnew = user_devices(phone_id=request.headers.get("phone_id"), id_user=result.id,
62                                               phone_model=request.headers.get("phone_model"), phone_brand=request.headers.get("phone_brand"))
63                     #Añade un nuevo registro en la tabla user_devices
64                     device = self.user_devicesRepository.add(userdevicesnew, db)
65                     return result, device
66                 else:
67                     #Comprueba si el dispositivo está bloqueado
68                     if device_db.block == True:
69                         return result, device_db, True
70                     return result, None, False
71             else:
72                 return None, None, False
```

Código 16 - Repositorio

PUESTA EN
MARCHA

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

9. PUESTA EN MARCHA

La puesta en producción es el proceso final en el ciclo de vida del desarrollo de software, donde una aplicación o sistema se despliega para su uso real por los usuarios finales. Este proceso incluye varias etapas críticas, como la preparación del entorno de producción, la migración de datos desde entornos de prueba, y la verificación de que todos los componentes funcionan correctamente en el entorno en vivo. Es esencial realizar pruebas exhaustivas y una revisión final para asegurar que no haya errores que puedan afectar el rendimiento o la funcionalidad del sistema.

Una vez que el sistema está en producción, se lleva a cabo un monitoreo constante para asegurar su estabilidad y desempeño óptimo. Esto implica la supervisión de los recursos del servidor, el rendimiento de la aplicación y la experiencia del usuario. Cualquier incidencia detectada debe ser gestionada rápidamente para minimizar el impacto en los usuarios. La puesta en producción debe incluir el soporte técnico inicial para resolver cualquier duda o problema que pueda surgir.

9.1. Securización

En la era digital actual, la seguridad de la información se ha convertido en una prioridad esencial para cualquier producto. A medida que las amenazas ciberneticas evolucionan y se vuelven más sofisticadas, es crucial implementar estrategias y tecnologías para proteger los datos y garantizar la integridad de los sistemas.

Elementos como los JWT ([JSON Web Tokens](#)), el algoritmo de hash MD5 y los firewalls de Azure juegan roles importantes en la construcción de un entorno seguro.

Cada uno de estos elementos aborda diferentes aspectos de la seguridad, desde la autenticación y autorización de usuarios hasta la verificación de integridad de datos y la protección contra ataques externos.



En los siguientes puntos, exploraremos cómo estas tecnologías contribuyen a fortalecer la seguridad.

- **JWT**

JSON Web Tokens ([JWT](#)) es un estándar para la creación de tokens de acceso que permiten la verificación de identidades entre diferentes partes. Se utiliza para la autenticación y autorización de usuarios. Un JWT contiene tres partes: el [header](#) (encabezado), el [payload](#) (carga útil) y la firma. El header especifica el algoritmo de cifrado utilizado, el payload contiene las declaraciones ([claims](#)) como la identidad del usuario y la expiración del token, y la firma verifica que el token no ha sido alterado.

FastAPI integra JWT a través de los [middleware](#) y [decoradores](#) para manejar sesiones de usuarios de manera segura y eficiente, permitiendo a los desarrolladores proteger rutas y controlar el acceso a los recursos según los permisos definidos.

```
● ● ●
1 from fastapi import Request
2 from config.jwt import validate_token
3 from fastapi.routing import APIRoute
4 from fastapi import HTTPException
5
6
7 class VerifyTokenRoute(APIRoute):
8
9     def get_route_handler(self):
10         original_route = super().get_route_handler()
11
12         @async def verify_token_middleware(request: Request):
13             try:
14                 token = request.headers["Authorization"].split(" ")[1]
15             except Exception:
16                 raise HTTPException(401, "No autorizado")
17
18             validation_response = validate_token(token, output=False)
19             if validation_response == None:
20                 return await original_route(request)
21             else:
22                 return validation_response
23
24         return verify_token_middleware
25
```

Código 17 - Middleware JWT

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- [MD5](#)

MD5 ([Message Digest Algorithm 5](#)) es una función criptográfica de hash que produce un resumen ([digest](#)) de 128 bits ([16 bytes](#)) a partir de una entrada de texto. Se utiliza comúnmente para verificar la integridad de archivos y almacenamiento de contraseñas.

- [Firewall de Azure](#)

El firewall de Azure es un servicio de seguridad en la nube que proporciona protección de red y controla el tráfico de entrada y salida en entornos de Azure. Este firewall nos posibilita establecer políticas para permitir o bloquear el tráfico basado en direcciones IP, puertos y protocolos. Azure Firewall se puede configurar para filtrar tráfico malicioso y prevenir ataques como inyecciones de SQL, DDOS y otros tipos de amenazas.

9.2.Legalidad

Para el proyecto Chikara se ha decidido utilizar la licencia [GNU General Public License Version 3 \(GPLv3\)](#) por varias razones estratégicas y filosóficas que se alinean con los objetivos y valores fundamentales de Café Con Palito. La GPLv3 es una licencia de software libre que garantiza a los usuarios la libertad de usar, estudiar, compartir y modificar el software. Esto es esencial para el proyecto Chikara, ya que busca fomentar la colaboración abierta y el desarrollo comunitario.

Una de las principales ventajas de la GPLv3 es que asegura que cualquier modificación o mejora del software también se distribuya bajo la misma licencia, lo que significa que las contribuciones de la comunidad seguirán siendo libres y abiertas. Esto no solo promueve la transparencia y la cooperación, sino que también ayuda a evitar que el software sea apropiado de manera privada, asegurando que siempre esté disponible para el beneficio de todos.

Además, la GPLv3 incluye disposiciones importantes para proteger a los usuarios contra patentes y restricciones legales que podrían limitar sus derechos a usar el software como deseen. Esto proporciona una capa adicional de seguridad y

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

confianza, tanto para los desarrolladores como para los usuarios finales, al garantizar que el software permanezca libre de obstáculos legales que puedan surgir en el futuro.

En resumen, la elección de la GPLv3 para el proyecto Chikara refleja un compromiso con los principios del software libre y la creencia en un ecosistema de desarrollo colaborativo y transparente. Al adoptar esta licencia, Café con Palito no solo protege su código, sino que también contribuye al crecimiento y la sustentabilidad de la comunidad del software libre.

9.3. De entorno DEV a PRO

Para realizar el despliegue del proyecto chikara hay que diferenciar el frontend y el backend ya que cada uno tiene un despliegue totalmente diferente.

El despliegue del frontend se hace generando un archivo APK que es subido a la Google Play Store con una licencia de desarrollador activa, tras pasar el proceso de seguridad de Google, la aplicación es publica y puede ser descargada por cualquier persona que disponga de un móvil con S.O Android 8.0 o superior.

Para el despliegue del backend hay que clonar todos los servicios del entorno dev que se mencionaron en el apartado [7.5 Despliegue](#). Si no se dispone de dicho entorno y se despliega directamente en el entorno pro ([Mala práctica de desarrollo](#)), hay que crear los microservicios como en el apartado [7.5 Despliegue](#) y posteriormente crear un archivo “`.env`” que contenga todas las variables de entorno necesarias para el correcto funcionamiento de la API Rest con la siguiente estructura:



```
1 # Variables de postgresql en azure
2 DB_HOST='chikaradb.postgres.database.azure.com'
3 DB_PORT='5432'
4 DB_NAME='chikara'
5 DB_USER='chikara'
6 DB_PASSWORD='y6c336n8b79AsA'
7
8 ## Variables de mongoDB
9 DB_USER_MONGO = 'cafecopalito'
10 DB_PASSWORD_MONGO = '1234'
11 DB_CLUSTER = 'clusterchikara.tzysdgw.mongodb.net'
12 DB_NAME_MONGO = 'ChikaraDB'
13 DB_COLLECTION = 'chiks'
14
15 ## Variable para JWT
16 KEY_cafecopalito
17 EXPIRE_TIME=1
18
19 ## Variables de correo
20 connection_string="endpoint=https://chikaracomunication.europe.communication.azure.com;accesskey=M0@lil0Cry43IgDT/TGD5TB9//Gr8ijbashNSVhxTcWxZxEpgYrkWPfgAiY1rtPsCSAR+ynvQ1D2lQYL/WrFBg=="
21 email="DoNotReply@f1fb054d5-0000-412d-ab6a-e3175252ba7.azurecomm.net"
22 url="https://chikara-backend.azurewebsites.net/"
23
24 # Variables Blob Storage
25 STORAGE_CONTAINER='chiks'
26 STORAGE_URL="https://chikarablobstorage.blob.core.windows.net/chiks"
27 STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;AccountName=chikarablobstorage;AccountKey=1tNvdkhFo/wHr1pADD/wQRmgskNPGjQTo5Gk7p6FX7NZdtelkr1ajQF+5XVYsUkXINAQkrH59Hk4+ASTMEpnVA==;EndpointSuffix=core.windows.net"
```

Código 19 - .env

Tras ello hay que generar una imagen de [Docker](#), para lo que hay que tener un archivo [Dockerfile](#) en la raíz del proyecto como el siguiente:

```
1 FROM python:3.11.4
2
3 WORKDIR /app
4 COPY ./requirements.txt /app/requirements.txt
5 RUN pip install --no-cache-dir -r /app/requirements.txt
6
7 COPY ./config /app/config
8 COPY ./decorators /app/decorators
9 COPY ./generator /app/generator
10 COPY ./Logs /app/logs
11 COPY ./middlewares /app/middlewares
12 COPY ./models /app/models
13 COPY ./repository /app/repository
14 COPY ./routers /app/routers
15 COPY ./schemas /app/schemas
16 COPY ./services /app/services
17 COPY ./temp /app/temp
18 COPY ./env /app/.env
19 COPY ./env.pro /app/.env.pro
20 COPY ./etl_chikara.py /app/etl_chikara.py
21
22 COPY ./main.py /app/main.py
23
24
25 EXPOSE 8000
26 CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Código 18 - Dockerfile



El siguiente paso es subir la imagen en el servicio [Azure Container Registry](#) para que la aplicación web sea capaz de ejecutar el contenedor que acabamos de generar, para ello hay que ejecutar los siguientes comandos en la terminal del sistema.

```
"docker build -t chikara:latest ."
```

```
"docker tag chikara:latest {URL_ACR}/chikara:latest"
```

```
"docker login {URL_ACR} -u {USER_ACR} -p {PWD_ACR}"
```

```
"docker push {URL_ACR}/chikara:latest"
```

Tras seguir los pasos el backend ya se encuentra desplegado y puede ser usado por cualquier dispositivo móvil que tenga la aplicación de chikara.

PLAN DE

PRUEBAS

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10. PLAN DE PRUEBAS

10.1. Test Unitarios

Los Test en Android se realizan en dos entornos distintos utilizando Test Unitarios y Test de Instrumentación. Ambos tipos de tests son esenciales para garantizar la calidad del código en una aplicación Android, ofreciendo diferentes niveles de validación, asegurando la calidad:

Los test unitarios en Android se utilizan para verificar el comportamiento de unidades individuales de código, como métodos o clases. En Android Studio, los test unitarios se escriben usando el framework JUnit y se ejecutan en la JVM local, sin necesidad de un dispositivo o emulador Android. Estos tests son rápidos de ejecutar y se utilizan para asegurar que cada parte del código funciona correctamente de manera aislada.

Los test de instrumentación se utilizan para probar el comportamiento del código en un entorno Android real o emulado. Pueden interactuar con los componentes del sistema operativo Android, como Activities, Services y Content Providers. Se utilizan para verificar la funcionalidad de la aplicación en un entorno completo y realista. Estas pruebas se realizan en un entorno aislado para no comprometer la integridad del dispositivo.



Test unitarios realizados:

Para las clases independientes que se utilizan para validar campos o cifrar información se generaron los test necesarios. Estos funcionaron correctamente.

```
class ValidatorFieldsTest {  
    lateinit var validateFields: ValidateFields  
    @Rule  
    @Before  
    fun setup() {  
        validateFields = ValidateFields()  
    }  
    /**  
     * test for a valid Username  
     * Correct username must have: at least 3 characters and not start with '@'  
     * @throws AssertionError  
     */  
    @Test  
    fun isValidUserName() {  
        val test = "username"  
        val result = Boolean::class.java.getMethod("validateUserName", String::class.java).invoke(validateFields, test)  
        assertTrue(result)  
    }  
}
```

Test Results: 10 ms Tests passed: 9 of 9 tests - 10 ms
Executing tasks: [:app:testDebugUnitTest, --tests, com.cafeconpalito.chikara.utils.ValidateFieldsTest] in project C:\Users\produccion\OneDrive\Documentos\AndroidStudioProjects\ChikaraAndroid

Captura 4 - Test Unitarios

Test de instrumentación:

Para realizar dichas pruebas fue necesario la inyección de dependencias antes de realizarlos, como se comentó en la descripción es necesario emular el entorno real de trabajo. Se realizaron test a los Endpoints de la Api obteniendo las respuestas esperadas de esta.

```
class RegisterUseCaseTest {  
    @Test  
    fun userNotExist() {  
        val test = "@thisUserNotExistsInTheDataBase"  
        val result = Boolean::class.java.getMethod("runBlocking", [Unit]).invoke(repository.userExist(test))  
        assertFalse(result)  
    }  
    /**  
     * Test for userNotExist  
     * Consult if given username is already registered in the System  
     * @throws AssertionError  
     */  
    @Test  
    fun userExist() {  
        val test = "@testUsername"  
        val result = Boolean::class.java.getMethod("runBlocking", [Unit]).invoke(repository.userExist(test))  
        assertTrue(result)  
    }  
}
```

Test Results: 6 passed 6 tests, 28s 530ms
Connected to process 4640 on device Pixel_3_XL_Prueta_Curso.
Starting a tests on Pixel_3_XL_Prueta_Curso(AVD) - 8.1.0
BUILD SUCCESSFUL in 2s
80 actionable tasks: 5 executed, 75 up-to-date
Build Analyzer results available

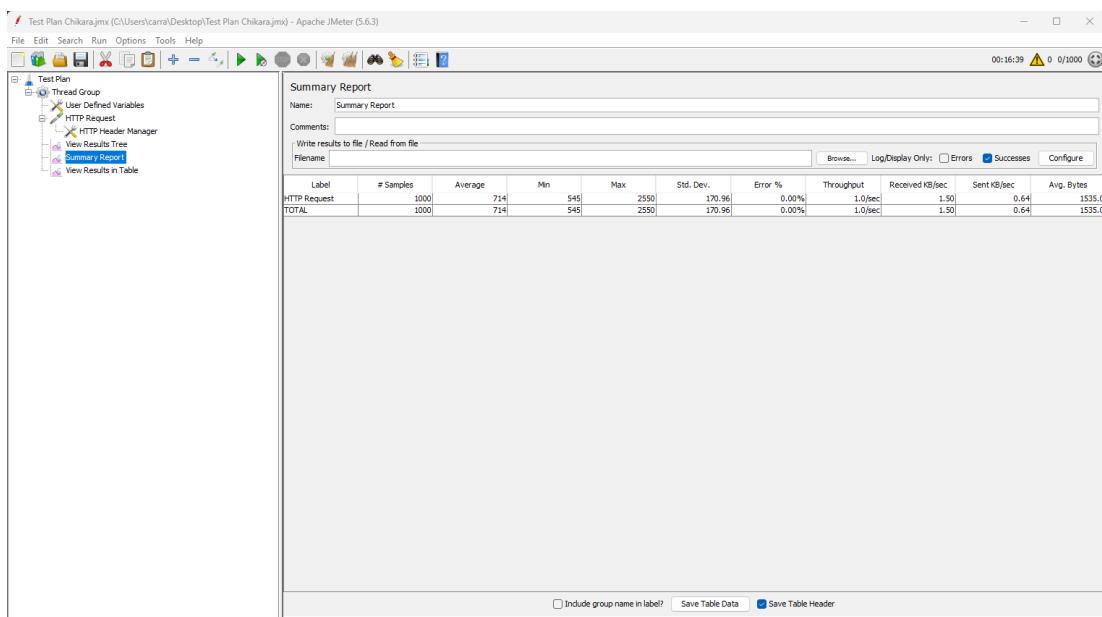
Captura 5 - Test Instrumentación

| | |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

10.2. Test de esfuerzo

Hacer pruebas de esfuerzo a un API REST es muy importante para evaluar su rendimiento y estabilidad bajo condiciones de alta carga. Estas pruebas permiten identificar cuellos de botella, verificar la escalabilidad, optimizar el uso de recursos y garantizar que el API pueda manejar el tráfico esperado. Al realizar estas pruebas, se previenen problemas críticos en producción, se mejora la experiencia del usuario, proporcionando un servicio robusto y fiable.

Para probar los límites del backend hemos realizado varias pruebas de esfuerzo, la primera es ver hasta cuantas peticiones podemos aceptar. Tras varias pruebas determinamos que el límite está en 1000, 1 por cada segundo, pero este límite puede ser superado con facilidad sin suponer un gran esfuerzo para el sistema.



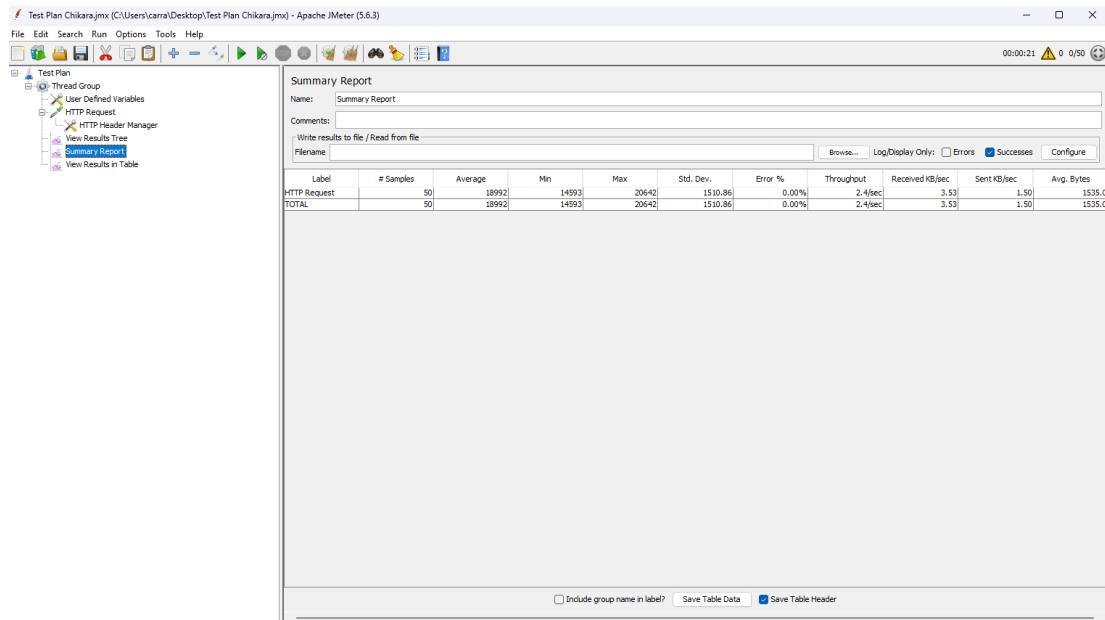
Captura 6 - JMeter Nº Máximo de Peticiones

El otro límite que queríamos encontrar es el de usuarios simultáneos, tras varias pruebas descubrimos que el límite está en 50 usuarios. Este límite está definido por el número de workers de Gunicorn ([Servidor donde se ejecuta el código de FastAPI](#)), que por defecto es 1.



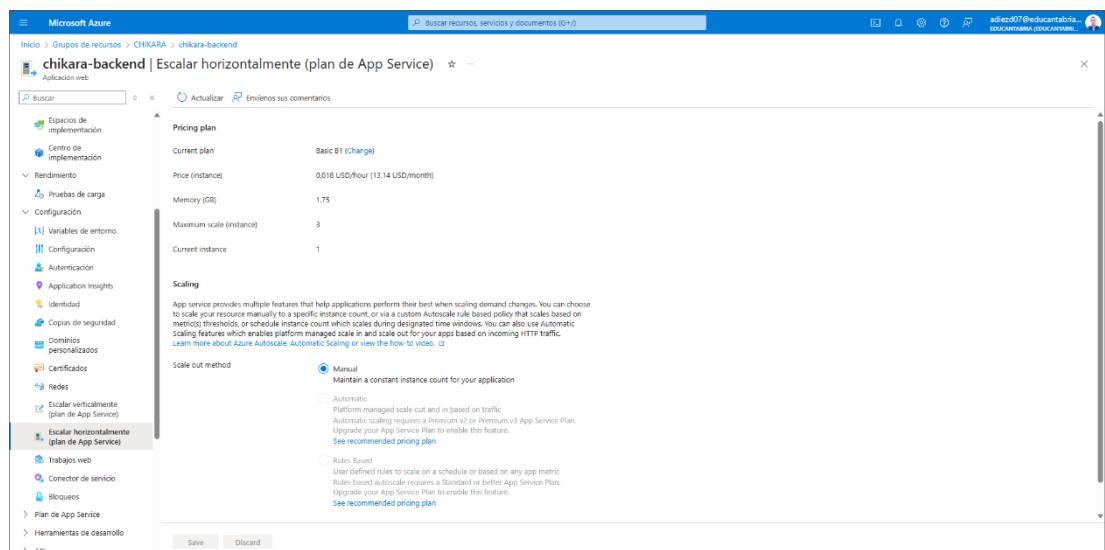
Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara



Captura 7 - JMeter Usuarios Simultáneos

Aunque los límites de chikara API son los ya definidos estos pueden ser aumentados cambiando el plan de Azure, o cambiando el número de workers de Gunicorn.



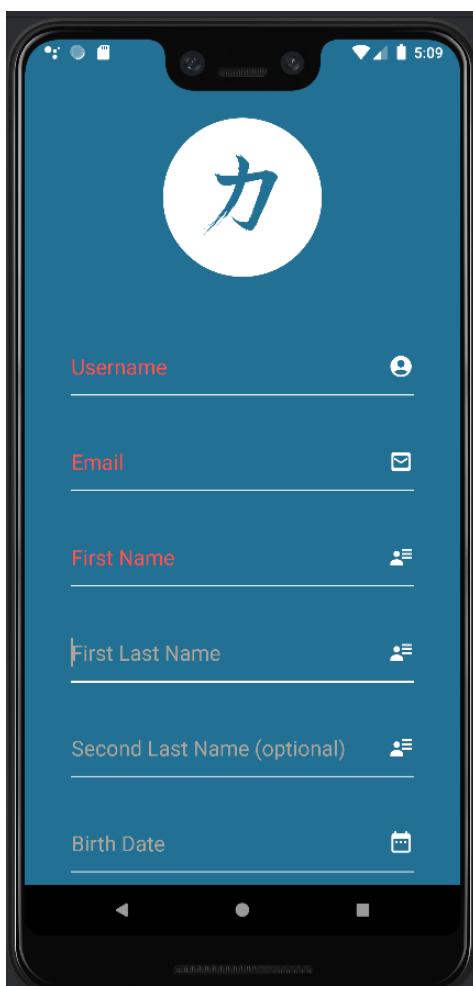
Captura 8 - Azure Plan App Web



10.3. Plan de Pruebas nivel Usuario

Tras las pruebas a nivel de código, hemos realizado pruebas directamente sobre la aplicación para comprobar si mantiene los estándares de calidad de Café Con Palito.

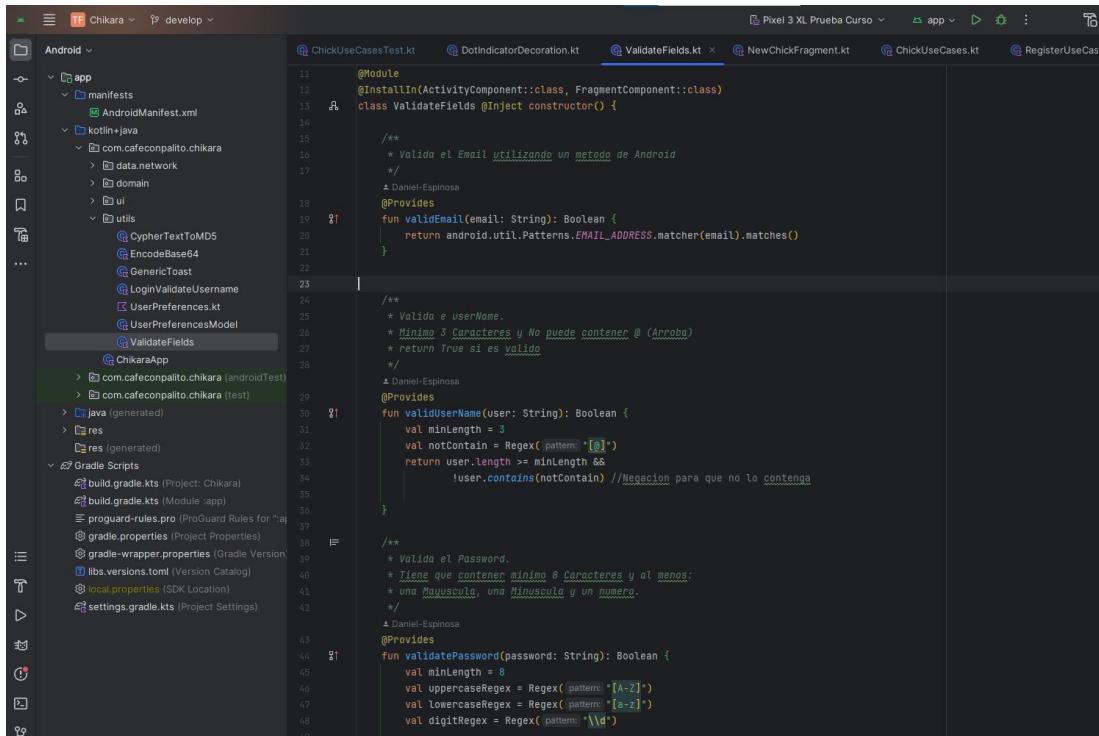
En el plan de pruebas se contempló que el usuario no pueda dejar campos sin llenar, mostrando en rojo los campos que no están llenos y no permitiendo realizar la acción si estos no están correctamente cumplimentados.



Captura 9 - Registro de Usuarios



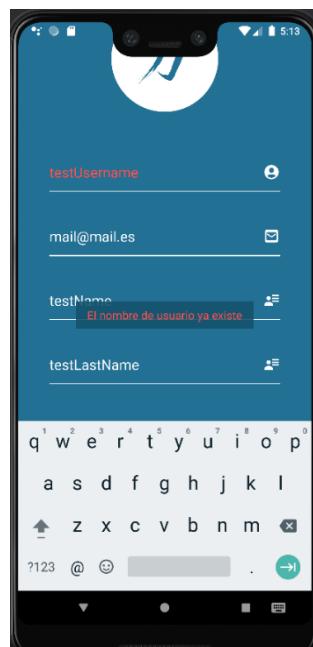
Para ello se creó una clase `ValidateFields` en el paquete `Utils` para unificar y simplificar las validaciones necesarias.



```
11  @Module
12  @InstallIn(ActivityComponent::class, FragmentComponent::class)
13  class ValidateFields @Inject constructor() {
14
15      /**
16       * Valida el Email utilizando un metodo de Android
17       */
18      @Provides
19      fun validateEmail(email: String): Boolean {
20          return android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()
21      }
22
23      /**
24       * Valida e userName.
25       * Tiene que contener minimo 3 Caracteres y No puede contener @ (Arriba)
26       * return True si es valido
27       */
28      @Provides
29      fun validateUserName(user: String): Boolean {
30          val minLength = 3
31          val notContain = Regex(pattern: "[@]*")
32          return user.length >= minLength &&
33                  !user.contains(notContain) //Negacion para que no lo contenga
34      }
35
36      /**
37       * Valida el Password.
38       * Tiene que contener minimo 8 Caracteres y al menos:
39       * una Mayuscula, una Minuscula y un numero.
40       */
41      @Provides
42      fun validatePassword(password: String): Boolean {
43          val minLength = 8
44          val uppercaseRegex = Regex(pattern: "[A-Z]+")
45          val lowercaseRegex = Regex(pattern: "[a-z]+")
46          val digitRegex = Regex(pattern: "\\d")
47
48      }
49
50  }
```

Código 20 - `ValidateFields`

También se utilizan **Toast** (*Mensajes flotantes en pantalla para avisar al usuario de errores*).



Captura 10 - `Toast`



Estos **Toast** los generamos desde la clase **GenericToast**, en el paquete Utils para controlar que el formato sea siempre igual.

```
companion object {

    /**
     * Generate a generic toast.
     * @param context Application Context
     * @param message String to display
     * @param toastLength Duration of Toast
     * @param isError type of Toast
     */
    new fun generateToast(
        context: Context,
        message: String,
        toastLength: Int,
        isError: Boolean
    ): Toast {

        val binding:CustomToastLayoutBinding = CustomToastLayoutBinding.inflate(LayoutInflater.from(context))

        //Setamos el mensaje
        binding.customToastText.text = message

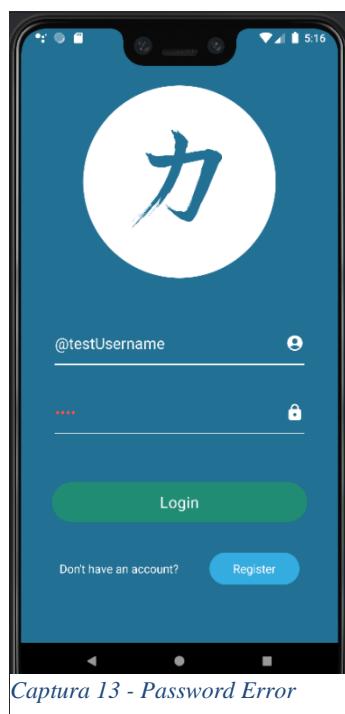
        //Setear el color del texto
        binding.customToastText.setTextColor(
            if (isError) ContextCompat.getColor(context, R.color.toast_text_error)
            else ContextCompat.getColor(context, R.color.toast_text)
        )

        binding.customToastText.setBackgroundResource(R.color.toast_background)

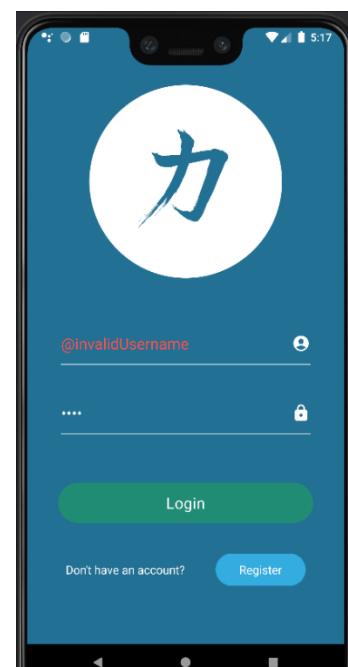
        //Creamos un objeto Toast
        val toast = Toast(context)
        toast.setGravity(Gravity.CENTER, xOffset: 0, yOffset: 0)
        toast.duration = toastLength
        toast.view = binding.root
        return toast
    }
}
```

Captura 11 - GenericToast

También advierte si, al realizar la acción, alguno de los campos no es correcto. Como por ejemplo en el **login**.



Captura 13 - Password Error



Captura 12 - Usename Error

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

10.4. Beta Testers

Hemos solicitado a un grupo de personas cercanas con un nivel de dominio de las tecnologías bastante dispar, para poder abarcar un amplio abanico de posibilidades, la prueba de la aplicación. El resultado viene a confirmar lo que hemos probado hasta ahora. Sin embargo, hemos recibido algún comentario que nos ha ayudado a mejorar cosas de las que no éramos conscientes.

Los reportes han sido los siguientes:

- Los márgenes del formulario de Nuevo chik no permiten que el usuario visualice el campo completo. **Corregido**.
- El nombre de usuario se guardó con un espacio de más y no me conseguía loguear. **Corregido**.
- El email de bloqueo de dispositivo no funciona correctamente. **Corregido**.
- No entiendo nada porque todo está en inglés. **No corregido**. **Fuera de alcance**.
- Varias faltas de ortografía (**spelling**), en diferentes puntos de la aplicación. **Corregido**.
- Los datos del usuario conectado vienen sin precargar. La información no aparece. **Corregido**.
- El visualizador de imágenes es demasiado pequeño. **Corregido (parcialmente)**.

PLAN DE

EMPRESA



11. PLAN DE EMPRESA

11.1. IDENTIFICACIÓN

11.1.1. Historia de la empresa

Todo comenzó en la máquina del café.

Gracias a la aleatoriedad con la que ésta suelta los palitos con los que se remueve, comenzamos a hablar entre nosotros. Muy pronto, cuando a Daniel se le ocurrió traer palitos de otra máquina diferente, comenzamos a ayudarnos los unos a otros. Siempre éramos los mismos. Siempre los componentes de la empresa actual.

Nos echábamos un cable en todo lo que podíamos y, sin darnos cuenta, lo que en un principio era una cuestión de compañerismo terminó convirtiéndose en una amistad.

Trabajamos muy bien juntos, porque compartimos muchos valores y, gracias a que los campos de los que procedemos son muy dispares, abarcamos todos los pilares que nuestra empresa necesita para funcionar.

Café con Palito tuvo su estreno triunfal en el segundo reto del primer curso del ciclo formativo DAM del curso 22/23, donde demostró ser una empresa eficaz y competitiva con unas calificaciones impresionantes (y merecidas), por un trabajo del que, aún en el segundo curso, seguimos rescatando código para nuevos trabajos.

11.1.2. Cultura empresarial

Nuestra misión es cubrir las necesidades presentes y futuras, que suponen los avances informáticos y tecnológicos para las pequeñas y medianas empresas, nuestra empresa se enfoca a facilitarle el tránsito aportando soluciones personales a cada una, para adaptarlo rápida y satisfactoriamente a los nuevos tiempos.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Nuestro campo de actividad se centra en el desarrollo de software e implantación de servicios / soluciones informáticas en un mercado local / regional centrado en pequeñas y medianas empresas.

La empresa Café con Palito comienza como una empresa de ámbito regional trabajando de forma local, captando clientes de forma directa, pero con pretensiones de convertirse en nacional, e incluso internacional, en un corto periodo de tiempo ya que el sector tecnológico, específicamente el desarrollo de software y soluciones digitales lo permite con relativa facilidad.

Empezaremos trabajando los 4 fundadores de la empresa y a medida que ganemos clientes, iremos contratando empleados que se alineen con los valores de nuestra empresa.

Nuestros valores los resumimos con una frase. **“Caminando hacia la Agenda 2030”**.

- **Innovación:**

Buscamos constantemente nuevas formas de crear experiencias multimedia únicas y emocionantes para nuestros clientes.

- **Creatividad:**

Fomentamos un entorno donde la creatividad florezca, permitiendo que cada miembro del equipo aporte ideas innovadoras.

- **Calidad:**

Nos comprometemos a ofrecer productos de la más alta calidad, asegurándonos de que cada aplicación multimedia cumpla con los estándares más exigentes.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **Colaboración:**

Creemos en el poder de la colaboración. Trabajamos estrechamente con nuestros clientes y dentro del equipo para lograr resultados excepcionales.

- **Adaptabilidad:**

Nos adaptamos rápidamente a los cambios tecnológicos y a las necesidades del mercado.

- **Transparencia:**

Fomentamos la honestidad y la transparencia en todas nuestras interacciones, tanto internas como externas.

- **Responsabilidad Social:**

Contribuimos positivamente a la sociedad mediante iniciativas de responsabilidad social empresarial y prácticas éticas.

- **Empatía:**

Nos preocupamos por comprender las necesidades y perspectivas de nuestros clientes, usuarios y colegas, actuando con empatía en todas las situaciones.

- **Inclusividad:**

Celebramos la diversidad y promovemos un ambiente inclusivo donde todas las voces son valoradas y respetadas.

- **Flexibilidad:**

Adoptamos una mentalidad flexible para adaptarnos a cambios rápidos y satisfacer las demandas del mercado y de nuestros clientes.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **Desarrollo Profesional:**

Apoyamos el crecimiento y desarrollo profesional de nuestro equipo, proporcionando oportunidades de aprendizaje y capacitación continua.

- **Inspiración:**

Buscamos inspirar a través de nuestro trabajo, generando experiencias multimedia que no solo cumplen funciones prácticas, sino que también generan emociones y experiencias memorables.

- **Compromiso con la Calidad de Vida:**

Reconocemos la importancia del equilibrio entre el trabajo y la vida personal, promoviendo un entorno que respeta el bienestar de nuestros empleados.

- **Inversión en la Comunidad:**

Contribuimos al desarrollo de las comunidades en las que operamos, apoyando proyectos locales y participando activamente en iniciativas comunitarias.

- **Agilidad:**

Adoptamos metodologías ágiles para la gestión de proyectos, permitiendo respuestas rápidas a cambios y entregas iterativas eficientes.

- **Pasión por la Tecnología:**

Fomentamos una pasión compartida por la tecnología, inspirando a nuestro equipo a mantenerse actualizado y entusiasmado con las últimas tendencias y avances.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

11.1.3. Identificación del equipo

El equipo de personas que conforman Café con Palito está conformado por:

Ramiro Gutiérrez Valverde:

- **Formación:**
 - Licenciado en Bellas Artes.
 - Diplomado en fotografía.
 - CFGS Desarrollo de Aplicaciones Multiplataforma
- **Experiencia:**
 - Profesor en talleres de pintura.
 - Profesor en plan regional PROA, de refuerzo educativo.
 - Vendedor
- **Inquietudes:**
 - Gran interés por el mundo de la tecnología, la imagen y el diseño.

Daniel Espinosa García:

- **Formación:**
 - Diseño gráfico y producción editorial
 - CFGS Desarrollo de Aplicaciones Multiplataforma
- **Experiencia:**
 - Encargado de un Centro de trabajo (imprenta), realizando funciones de gestión del personal, producción, administración, trato con el cliente, suministros de materias primas para la empresa y mantenimiento.
- **Inquietudes:**
 - Interés en el desarrollo de aptitudes para la gestión de equipos de trabajo, enfocado en el ámbito tecnológico.
 - Inteligencia Artificial y su aplicación en productos y servicios.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Albano Diez de Paulino:

- **Formación:**
 - CFGS Mantenimiento Electrónico
 - CFGS Desarrollo de Aplicaciones Multiplataforma
- **Experiencia:**
 - Instalador de fibra FTTH
 - Técnico de Sonido
- **Inquietudes:**
 - Electrónica Analógica y Digital (Diseñar circuitos Electrónicos)
 - Informática (Tocar el Hardware de los pc etc. Y probar software nuevo).

11.1.4. Descripción actividad empresarial

Debido a la digitalización de las empresas y la tecnificación de estas, surge la necesidad de estas de adaptar y mejorar sus procesos y servicios para brindar un mejor producto.

Es, en este punto, en donde la empresa Café con Palito entra en juego para brindar a las empresas servicios de desarrollo de aplicaciones. Partiendo de algo tan sencillo como la creación de una página Web, la implementación y personalización de un ERP, servicio de venta online o cualquier necesidad que las empresas puedan tener.

Nombre del negocio:

Café Con Palito S.L.

- **Logotipo**



Desarrollo de Aplicaciones

Ilustración 1 - Logotipo CCP

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

○ **Ámbito de actuación:**

La empresa Café con Palito comienza como una empresa regional trabajando localmente, captando clientes directamente, pero con pretensiones de convertirse en nacional e internacional, en un corto periodo, ya que el sector tecnológico, específicamente el desarrollo de software y soluciones digitales lo permite fácilmente.

11.2. ANALISIS DE MERCADO

11.2.1. Análisis Macroentorno

El macroentorno es un conjunto de factores externos que afectan a una empresa o a un país en su conjunto. Vamos a explorar algunos aspectos relevantes del macroentorno en Europa y España:

Economía y Demografía de España:

- **Población:** España cuenta con una población de 48,327,927 personas y una moderada densidad de población de 96 habitantes por km².
- **PIB:** La economía española ocupa el 14 lugar en volumen de PIB.
- **Deuda Pública:** En 2022, la deuda pública española fue de 1,502,804 millones de euros, representando el 111.6% del PIB. Esto la sitúa entre los países con mayor deuda con relación al PIB a nivel mundial.
- **IPC:** La última tasa de variación anual del IPC en España, publicada en enero de 2024, fue del 3.4%.
- **Desempleo:** España se destaca por tener una de las tasas de desempleo más altas del mundo.
- **Calidad de Vida:** El PIB per cápita en 2023 fue de 30,320 euros, ubicando a España en el puesto 37 en el ranking mundial.
- **Índice de Desarrollo Humano (IDH):** Los españoles disfrutan de una buena calidad de vida según este índice de las Naciones Unidas.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **Facilidad para Hacer Negocios:** España ocupa el 31º lugar en el ranking “Doing Business”, que evalúa la facilidad para hacer negocios en diferentes países.
- **Corrupción:** El Índice de Percepción de la Corrupción en el sector público español es de 60 puntos, situándose en el puesto 35 a nivel global.

Contexto Político y Social:

- España es un estado social y democrático de derecho que valora la libertad, la justicia, la igualdad y el pluralismo político.
- El sistema europeo tal como se concibió hace 30 años presenta desafíos para enfrentar el nuevo contexto geopolítico global e interno de la UE
- La digitalización de empresas en España está en proceso y muestra efectividad, aunque lenta

Perspectivas y Desafíos:

España debe adaptarse a los cambios globales y aprovechar las oportunidades en un mundo en constante evolución.

En resumen, España enfrenta desafíos económicos, sociales y políticos, pero también tiene oportunidades para crecer y prosperar en el contexto europeo.

11.2.2. Datos generales del sector

El sector de consultorías informáticas en Cantabria presenta características interesantes. A continuación, analizaremos algunos aspectos relevantes:



Empresas Destacadas:

- **LIS Data Solutions S.L:** Es una empresa especializada en ciencia de datos y analítica avanzada. Su enfoque se centra en la transformación de negocios mediante el uso de tecnologías como inteligencia artificial, machine learning, big data y business intelligence. Trabajan con clientes para aprovechar al máximo el potencial de sus datos, ofreciendo soluciones en áreas como consultoría de Big Data, automatización de procesos, mantenimiento predictivo y más. Su objetivo es ayudar a las empresas a crecer y ser más competitivas a través del poder de los datos.
- **LKS:** Es una empresa de consultoría y servicios profesionales con una amplia trayectoria en ingeniería, tecnología y gestión empresarial. Su enfoque se centra en soluciones integrales para empresas, abarcando áreas como ingeniería industrial, tecnologías de la información, gestión de proyectos, sostenibilidad y transformación digital. LKS trabaja con clientes de diversos sectores para impulsar la innovación, la eficiencia y el crecimiento sostenible. Su compromiso es contribuir al éxito y desarrollo de las organizaciones mediante un enfoque colaborativo y orientado a resultados.
- **Acicatech:** Es una empresa que ayuda a las PYMES a modelar su estrategia y procesos en el mundo digital. Se especializan en ciencia de datos, analítica avanzada y tecnologías como inteligencia artificial, machine learning y big data. Trabajan con clientes para aprovechar al máximo el potencial de sus datos, ofreciendo soluciones en áreas como consultoría de Big Data, automatización de procesos y mantenimiento predictivo. Además, colaboran con partners y fabricantes de tecnologías de máxima implantación.
- **CIC Consulting Informático:** Esta empresa se dedica a la ingeniería y desarrollo de proyectos de informática y comunicaciones. Su enfoque está en la tecnología y la innovación.

| |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| Título: Proyecto Chikara |

Demanda de Profesionales IT y TIC:

- Los profesionales IT y TIC son altamente demandados en estos sectores. Los programadores y técnicos informáticos representan casi el 50% de las contrataciones.
- Existe una gran cantidad de puestos disponibles en el mercado laboral en comparación con el número de estudiantes y profesionales en el campo.

Digitalización y Desarrollo:

- Las empresas de consultoría informática en Cantabria contribuyen al mantenimiento y desarrollo de negocios mediante soluciones tecnológicas.
- Ofrecen servicios de asesoramiento, desarrollo web, asistencia técnica y soluciones de gestión.

Ayudas y Subvenciones:

- El Kit Digital de Acelera Pyme ofrece ayudas para la digitalización de empresas. Las consultoras pueden ayudar a analizar las necesidades y solicitar estas subvenciones.

En resumen, el sector de consultorías informáticas en Cantabria está en constante crecimiento y ofrece oportunidades para profesionales y empresas que buscan innovar y mejorar su presencia digital.

11.2.3. Clientes Potenciales

En un mundo cada vez más conectado, la digitalización se ha convertido en un pilar fundamental para el éxito de los negocios, independientemente de su tamaño. Sin embargo, es cierto que las grandes consultoras a menudo se enfocan en atender a las corporaciones y dejan de lado a los comercios pequeños. Aquí Café con Palito encuentra su oportunidad.

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

Café con Palito, con su enfoque en el mercado local, está perfectamente posicionado para aprovechar este nicho. Veamos cómo:

Personalización y Cercanía:

- Los comercios pequeños tienen la ventaja de conocer a sus clientes de manera más cercana. Café con Palito puede aprovechar esta relación personalizada para ofrecer soluciones digitales adaptadas a las necesidades específicas de cada cliente.
- La digitalización no se trata solo de tecnología, sino también de conectar con las personas. Aquí es donde los comercios pequeños pueden brillar.

Servicios a Medida:

- Mientras las grandes consultoras ofrecen soluciones genéricas, Café con Palito puede diseñar estrategias digitales específicas para cada negocio pequeño.
- Desde la creación de una página web sencilla hasta la gestión de redes sociales o la implementación de sistemas de pedidos en línea, la adaptabilidad es la clave.

Creatividad y Flexibilidad:

- Los comercios pequeños pueden ser más ágiles en la implementación de nuevas ideas. Café con Palito puede explorar opciones creativas, como programas de fidelización digital, marketing local o incluso colaboraciones con otros negocios cercanos.
- La flexibilidad para adaptarse a las cambiantes tendencias digitales es un activo valioso.

Educación y Acompañamiento:

- Muchos comerciantes pequeños pueden sentirse abrumados por la digitalización. Café con Palito puede ofrecer talleres, asesoramiento y acompañamiento en todo el proceso.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- La educación sobre herramientas digitales y su aplicación práctica es esencial para el éxito.

En resumen, Café con Palito tiene la oportunidad de ser un aliado estratégico para los comercios pequeños en su camino hacia la digitalización. ¡Que el aroma del café se mezcle con el mundo digital y cree experiencias únicas para los clientes!

11.2.4. Plan de Marketing

Para fortalecer nuestra presencia y reconocimiento en el mercado, hemos diseñado una estrategia integral que abarca diversas actividades promocionales y de participación comunitaria.

- **Promoción en RRSS** publicación semanal del trabajo realizado
- **Participación en eventos sociales**, como carreras populares.
- **Participación en eventos tecnológicos y empresariales**, así como participar en eventos de la Cámara de comercio, etc.
- **Realización de charlas** en lugares educativos y del sector tecnológico.
- **Regalar merchandising** de nuestra empresa en los eventos a los que acudamos.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

11.2.5. DAFO

| DEBILIDADES | AMENAZAS |
|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Inexperiencia • Falta de capital • Miedo al emprendimiento | <ul style="list-style-type: none"> • Competencia • Irrupción de las IAs |
| FORTALEZAS | OPORTUNIDADES |
| <ul style="list-style-type: none"> • Formación • Motivación • Integración en la sociedad | <ul style="list-style-type: none"> • Mercado en continua expansión • Aumento de posibles clientes ante la implementación de las nuevas medidas tecnológicas en España |

11.3. PLAN TECNICO-PRODUCTIVO

11.3.1. Servicios Ofrecidos

Como hemos señalado antes en el punto [11.2.3 Clientes Potenciales](#), Café con Palito ofrece a sus clientes el desarrollo de software exclusivo, o adaptado a sus necesidades y la implantación de soluciones informáticas a los diferentes problemas y/o retos que puedan enfrentar, facilitando su trabajo y el desempeño de sus actividades.

11.3.2. Tabla de Precios

Cada proyecto que desarrollamos tiene un coste variable en función de distintos factores, como el tiempo necesario para el desarrollo del producto, la cantidad de personas que trabajarán en el mismo y las licencias necesarias que haya que utilizar para su entrega, entre otros. Es por ello por lo que hablar de una tarificación estándar o una tabla de precios no es algo que tenga mucho sentido. Antes de comenzar un proyecto se realiza un estudio pormenorizado que incluye un precio convenido con el cliente que ha de ser aceptado antes de comenzar a trabajar.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

11.3.3. Distribución

La distribución de nuestros productos también es particular, puesto que puede variar desde una entrega virtual de un servicio que funcione vía web, como puede ser una aplicación o un programa descargable y autoinstalable, o una entrega in situ que requiera de una instalación personalizada con elementos físicos como un servidor.

11.4. PLAN DE RECURSOS HUMANOS

11.4.1. Perfiles profesionales

Buscamos personas con experiencia en las siguientes áreas:

- **Desarrollo de software:** dominio de lenguajes de programación como Python, Java, C#, etc., conocimientos de bases de datos, frameworks, APIs, testing, etc.
- **Marketing digital:** manejo de herramientas de SEO, SEM, redes sociales, email marketing, analítica web, etc., capacidad de crear y gestionar campañas, contenidos y estrategias digitales.

11.4.2. Organigrama

- **CEO** | Daniel Espinosa García
- **Administración y Finanzas** | María Carmen Barrios Fernández
- **Programmers Backend Senior Manager** | Albano Díez de Paulino
- **Backend Developer** | Ramiro Gutiérrez Valverde
- **Level Design and Art Senior Manager** | Ramiro Gutiérrez Valverde
- **Marketing** | María Carmen Barrios Fernández
- **Café Con Palito Academy** | Adrián Peña Carnero

11.4.3. Política Salarial

Valoramos el talento y el esfuerzo de nuestros empleados. Por eso, ofrecemos un salario competitivo en el sector, que se complementa con un plus extraordinario por rendimiento. Además, queremos que nuestros empleados se sientan motivados y satisfechos, por lo que ofrecemos beneficios sociales como

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

clases de inglés gratuitas. Y para garantizar que nuestros salarios se ajustan al mercado y a las necesidades de nuestros empleados, cada trimestre realizaremos reuniones para actualizarlos.

11.4.4. Política de Teletrabajo

Apostamos por la flexibilidad y la conciliación de la vida laboral y personal de nuestros empleados. Por eso, les ofrecemos la posibilidad de realizar el 30 % de su jornada laboral en teletrabajo, siempre que sus tareas se puedan realizar a distancia y no requieran su presencia en la oficina. Además, nuestros empleados pueden elegir el lugar desde el que quieren teletrabajar, siempre que lo comuniquen previamente a la dirección y que no sea un espacio público. Así, podrán disponer de dos lugares diferentes para trabajar desde casa, adaptándose a sus necesidades y preferencias.

11.4.5. Política acoso Laboral

En nuestra empresa, mantenemos un firme compromiso con la creación de un entorno laboral seguro, respetuoso y productivo para todos nuestros empleados. Para ello, hemos implementado una política rigurosa contra el acoso laboral. Creemos que cada individuo tiene derecho a trabajar en un ambiente libre de hostigamiento, intimidación y discriminación. Esta política no solo protege a nuestros empleados de conductas inapropiadas, sino que también promueve la igualdad, el respeto mutuo y el bienestar general. Al abordar y prevenir el acoso laboral, reforzamos nuestros valores corporativos y aseguramos que todos los miembros de nuestro equipo puedan desempeñarse al máximo de sus capacidades, contribuyendo así al éxito y la armonía de la empresa.



11.5. PLAN DE INVERSIONES

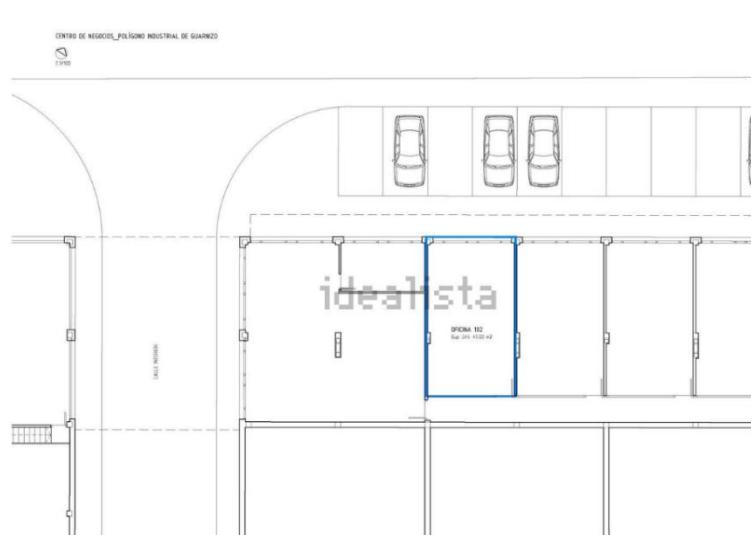
11.5.1. Localización Sede

Estará ubicado en el polígono Industrial de Guarnizo (El Astillero) Barrio La Estación N°72-A, Oficina 102. El Local cuenta con 48 m² diáfano, con un alquiler de 325 €/mes. Esta propuesta es de inicio con la posibilidad de ampliar otras oficinas adicionales en el momento que las necesitemos.

[Alquiler de Oficina en LA ESTACION, 72 A, Guarnizo — idealista](#)



Captura 14 - Anuncio Idealista - Fachada

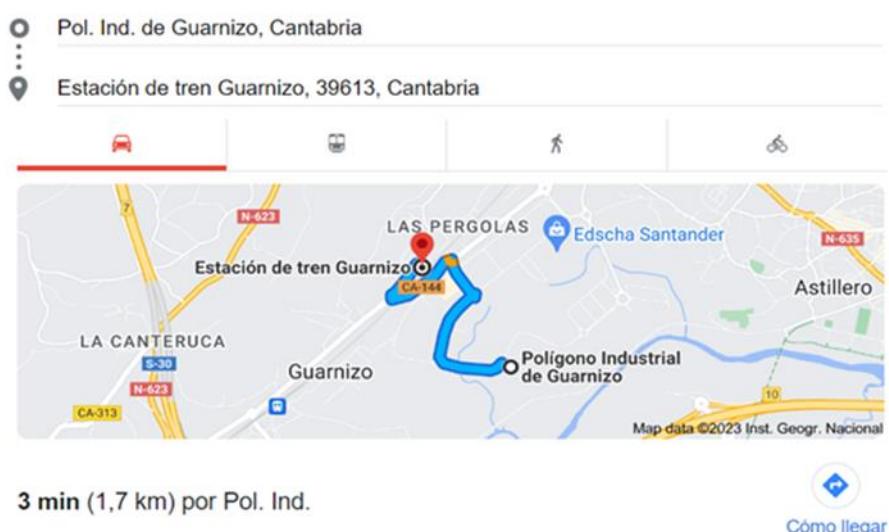


Captura 15 - Anuncio Idealista - Plano



Lo primero que nos atrajo del sitio son las futuras posibilidades de establecer contacto con potenciales clientes, al estar situado en el “Polígono industrial de Guarnizo” y la ampliación del “Polígono industrial de Morero” supone estar situado muy cerca de potenciales clientes que podamos captar. Además de contar con muchas plazas de aparcamiento.

Su comunicación es muy buena ya que está la estación de tren muy cerca.



Captura 16 - Google Maps

Por carretera está rodeada de distintos accesos:

- La N-623 está a unos 2 Km.
- La autovía A-8 a unos 3 Km.
- Y la N-634 a unos 3 Km.

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

11.5.2. Inmovilizado intangible.

| NOMBRE | AÑO 1º | AÑO 2º |
|-----------------|----------------|----------------|
| IntelliJ IDEA | 599.00€ | 479.00€ |
| Dbeaver | 250.00€ | 250.00€ |
| TOTALES: | 849.00€ | 729.00€ |

11.5.3. Inmovilizado Material

| NOMBRE | AÑO 1º | AÑO 2º |
|----------------------------------|------------------|-----------|
| Mesa escritorio FLEXISPOT E1 x 6 | 1.679,88€ | 0€ |
| Silla de Oficina ERGOCITY x 6 | 839,94€ | 0€ |
| Cafetera Nespresso | 69€ | 0€ |
| TOTALES: | 2.588,82€ | 0€ |

11.5.4. Presupuesto Detallado de Gastos

| NOMBRE | AÑO 1º | AÑO 2º |
|-------------------------------------|----------------------|-----------------------------------------|
| Gastos constitución | 350€ | 0€ |
| Alquiler x 12 | 3.900€ | 3.900€ sin contar con la subida del IPC |
| Luz con una previsión de 80€ al mes | 960€ aproximadamente | 960€ aproximadamente |
| Material de oficina | 350€ aproximadamente | 350€ aproximadamente |
| Renting PCs | 600€ | 600€ |
| Capsulas de café y leche | 350€ | 350€ |
| TOTALES: | 6.510€ | 6.160€ |

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

11.5.5. Resumen de Inversiones

| NOMBRE | AÑO 1º | AÑO 2º |
|-------------|-----------|----------|
| Inversiones | 3.437.82€ | 729€ |
| Gastos | 6.510€ | 6.160€ |
| Liquidez | 1000.00€ | 1000.00€ |

11.5.6. Fuentes de financiación

Para este proyecto, vamos a tener 3 diferentes fuentes de financiamiento:

- A. La primera de ellas será un [crowdfunding](#) usando la plataforma [Kickstarter](#).
- B. Nuestra segunda fuente de financiación va a ser las FFF ([Friends, Family and Fools](#)) siendo en nuestro caso la F de familia, es decir, que contaremos con una ayuda económica de nuestros familiares. Por lo general se considera a las FFF una fuente de financiación ajena, es decir, que habría que devolver el dinero, pero para nosotros va a ser como una fuente propia, casi como si fuese una aportación de capital de socios.
- C. Además de estas fuentes de financiación, también vamos a tratar de solicitar la ayuda convocada por la sociedad para el [Desarrollo de Cantabria SODERCAN](#), así como programa de préstamos participativos y participación en capital [Start Up Capital Cantabria](#).

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

11.6. OBLIGACIONES EMPRESARIALES

11.6.1. Forma jurídica

Como una empresa de reciente creación, hemos tomado la decisión de constituirnos como una Sociedad Limitada, asegurando una estructura legal que nos brinde flexibilidad y protección. Hemos establecido un capital inicial de 3.000€, lo cual nos permite comenzar nuestras operaciones con una base financiera sólida y alineada con nuestras expectativas de crecimiento. Esta forma jurídica no solo nos proporciona seguridad y confianza a nuestros socios, sino que también facilita nuestra capacidad para adaptarnos y expandirnos en el competitivo mercado actual.

11.6.2. Ventajas

- Menores costes de constitución, lo que nos resulta más asequible.
- Mayor sencillez de los trámites burocráticos tanto para su constitución como para su mantenimiento.
- Otra de las ventajas es que tiene opciones de desgravación, puede llegar a ser el 100% de los gastos en relación con su actividad económica. Además, a nivel fiscal, tiene posibilidades de tener descuentos tributarios según su nivel de beneficios.
- En una Sociedad Limitada si hay algún problema en la empresa, caso de deudas o problemas financieros, los socios no somos personalmente responsables con nuestro patrimonio personal, sino que la responsabilidad queda limitada al capital aportado a la empresa.
- No hay un máximo de integrantes de socios y podemos cobrar un salario cada uno. Por lo tanto, una manera de declarar menos a hacienda es redistribuir parte de los beneficios entre el sueldo de los socios.
- Otro punto a su favor es la facilidad que hay para acceder a la financiación, lo cual significa que hay productos específicos para ellos en los bancos.

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

- **Flexibilidad en la estructura:** esta opción permite una mayor flexibilidad en la estructura de la empresa. Los socios podemos acordar las condiciones y normas de funcionamiento de la sociedad a través de los estatutos sociales, esto facilita la adaptación de la empresa a las necesidades y circunstancias específicas nuestras.
- Consideramos que la forma jurídica de la sociedad limitada **es atractiva para los inversores futuros**, ya que ofrece un marco legal y de responsabilidad claro y definido. Además, esto facilita la obtención de financiación externa y la atracción de nuevos socios o inversores a la empresa.
- **Continuidad de la empresa:** la empresa puede seguir operando incluso si uno o varios socios se retiran o fallecen. Esto asegura la continuidad del negocio y evita posibles complicaciones en caso de cambios en la composición accionarial.

11.6.3. Desventajas

- **Mayor complejidad administrativa:** Las sociedades limitadas suelen requerir una mayor cantidad de trámites y formalidades administrativas en comparación con otras formas jurídicas más sencillas. Esto implica un mayor costo y tiempo dedicado a la gestión administrativa de la empresa.
- **Limitaciones en la toma de decisiones:** En una sociedad limitada, las decisiones importantes deben ser tomadas por los socios en asamblea general o mediante acuerdos unánimes. Esto puede llevar a una mayor burocracia y dificultar la toma de decisiones rápidas en situaciones que requieren agilidad y flexibilidad.
- **Capital social mínimo requerido:** Para constituir una sociedad limitada, se requiere un capital social mínimo establecido por la legislación. Este requisito puede suponer una barrera para emprendedores con recursos financieros limitados.

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

11.6.4. Pasos que rellenar para formar la empresa

- I. **Registro Mercantil Central.** Solicitud del certificado de nombre en nuestro caso “Café con palito”.
- II. **Cuenta bancaria** a nombre de la sociedad “Café con Palito, SL” donde cada socio ha desembolsado 750 €.
- III. **Estatutos Sociales.**
 - A. **Objeto Social.**

El diseño, comercialización, venta y cesión de uso del software. La intermediación y comercialización de servicios relacionados con el software de todo tipo de empresas.

B. Denominación Social.

“Café Con Palito, S.L.”

IV. Fecha cierre ejercicio.

Desde la fecha de firma de la escritura notarial hasta el 31 de diciembre de ese año corresponde al primer ejercicio. El resto de los años serán el año fiscal completo.

V. Participaciones/valor nominal.

El capital social se fija en 3000 € y está dividido en 300 participaciones sociales con un valor nominal, cada una de ellas, de 10 €. Cada socio adjudica 75 participaciones sociales.

VI. Sistema de Administración.

Hemos optado por 4 administradores con facultades solidarias:

- María Carmen Barrios Fernández
- Daniel Espinosa González
- Ramiro Gutiérrez Valverde
- Albano Diez de Paulino

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

VII. Escritura Pública- En la Notaría.

- A. Certificado remitido por el Registro Mercantil Central del nombre que nos ha remitido el Registro Mercantil.
- B. Estatutos Sociales.
- C. DNI/NIF original de todos los socios.
- D. Los 4 certificados bancarios, uno por cada socio fundador.

VIII. NIF provisional.

Rellenar Modelo 036.

IX. Alta IAE:

Rellenar Modelo 840: Está exento hasta 1.000.000€.

X. Declaración Censal para el IVA:

Rellenar Modelo 036:

XI. Registro Mercantil:

- A. Copia NIF provisional.
 - B. Copia escritura pública.
1. Obtención NIF definitivo.

11.6.5. Obligaciones Laborales

- Las empresas tienen varias obligaciones con sus trabajadores asalariados para garantizar un entorno laboral seguro y cumplir con la legislación. Aquí están algunas de las principales:
- Contratación de una mutua.
- Dar de alta en el régimen general de cotización a la Seguridad Social.
- Control Horario / Registro de Jornada: Desde mayo de 2019, es obligatorio registrar la jornada laboral para todos los trabajadores. Esto ayuda a controlar los tiempos de trabajo, el absentismo y las horas extras. Las empresas deben mantener registros diarios de la jornada, incluyendo el horario de inicio y finalización de cada empleado.

| |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| Título: Proyecto Chikara |

- Prevención de Riesgos Laborales y Reconocimiento Médico: Toda empresa con personal contratado debe cumplir con la Ley de Prevención de Riesgos Laborales. Esto implica evaluar los riesgos de la actividad, informar y formar a los trabajadores sobre estos riesgos, y vigilar su salud y aptitud para el trabajo.

11.6.6. Obligaciones sobre prevención Laboral

Nuestro lugar de trabajo cumple con la normativa establecida en el Real Decreto 486/1997 de 14 de abril, en el que se establecen las condiciones mínimas de seguridad y salud en los lugares de trabajo. Estas son las medidas más destacables que hemos tenido en cuenta:

- **Suelos:**

El pavimento del suelo elegido para su colocación es antideslizante y se ha evitado las pendientes y las rampas excesivas.

- **Temperatura:**

En nuestro lugar de trabajo cumplimos la normativa sobre temperaturas mínimas que establece que para las oficinas debe estar entre 17º y 27º C, con una adecuada ventilación natural. Tenemos puntos de dispensación de agua y diferentes líquidos para prevenir la deshidratación.

- **Iluminación:**

Contamos con un amplio ventanal que ofrece una apropiada iluminación natural y para cumplir holgadamente con la normativa, al tratarse de un lugar con exigencias visuales bajas, disponemos de una iluminación de 300 lux., apropiada incluso para exigencias visuales medias.

- **Riesgo Eléctrico:**

En nuestra empresa el único riesgo eléctrico posible corresponde a un contacto indirecto salvo la manipulación indebida de algún equipo electrónico (de lo que todo empleado está advertido e informado). Salvamos dicho riesgo



con una adecuada instalación eléctrica con tomas de tierra e interruptores diferenciales. El cuadro eléctrico está debidamente señalizado, como lo están los equipos que no deben ser manipulados bajo ningún concepto.

- **Incendios:**

Debido a que estamos trabajando con equipos electrónicos existe un riesgo de incendio eléctrico por lo que disponemos de extintores debidamente señalizados de CO2 y de polvo seco de clase “C”.

- **Salidas de evacuación:**

En el edificio se dispone de las salidas de evacuación completamente despejadas, señalizadas y con iluminación de seguridad. Las puertas se pueden abrir hacia afuera.

- **Discapacitados:**

Nuestras dependencias cumplen con las adecuadas medidas tendentes a facilitar la movilidad, accesibilidad y no discriminación de las personas con discapacidad para el acceso y utilización de nuestra oficina, suprimiendo todo tipo de barreras arquitectónicas, en particular las puertas, vías de circulación, escaleras, servicios higiénicos y puestos de trabajo (Siempre, claro, en la medida de lo posible, teniendo en cuenta la arquitectura del local).

CONCLUSIONES



12. CONCLUSIONES

A continuación, se presentan las opiniones de los integrantes de Café con Palito sobre el proyecto Chikara y el módulo 0492 - proyecto de desarrollo de aplicaciones multiplataforma. Estos comentarios reflejan las diversas perspectivas y experiencias de los miembros del equipo, brindando una visión integral sobre los desafíos y logros alcanzados durante la implementación y desarrollo del proyecto. Las opiniones abarcan desde aspectos técnicos hasta consideraciones personales, ofreciendo una comprensión profunda de cómo el equipo ha interactuado y evolucionado a lo largo del proceso.

Ramiro Gutiérrez Valverde:



Pese a la novedad del primer curso, y el hecho de que partiease prácticamente sin experiencia en el sector del desarrollo este año ha sido, con diferencia, en el que más he aprendido.

Todo adquiere sentido a medida que nos vamos acercando al final del curso. Las piezas por fin encajan sin fricción. Eres capaz de realizar tareas y proyectos que hace un año no cabían en tu imaginación y te sientes encantado.

Con ilusión afrontas el reto de adentrarte en el mundo laboral con las FCTs, pensando que estás preparado, pero te asustas porque te das cuenta de que aún tienes mucho que aprender.

He aprendido mucho este año, sí. Pero tal vez, de las cosas más importantes que he conseguido interiorizar haya sido el hecho de que, aunque no lo sepa todo, en la estructura, la base que he conseguido dibujar en mi cabeza, todo encaja.

Lo he podido comprobar al realizar este proyecto. Hemos escogido Android Studio y Kotlin para la parte del cliente y Python para la parte del servidor, lenguajes



que no hemos estudiado durante el curso, y hemos sabido adaptar nuestros conocimientos sin sentir que partíamos de cero. Esa es la confianza que necesita un profesional. Eso es lo que más valoro de todo lo que he aprendido. Los profesores nos han dirigido muy sabiamente para fortalecer esa confianza, esa autonomía a la hora de aprender, que es la base de todo proceso de aprendizaje.

Las FCTs, como decía, me han hecho ahondar en la sensación de que, pese a que este mundo es muy grande, puedo comprender y afrontar todo lo que me proponga. El centro de trabajo me ha recibido con mucha atención y me he sentido muy arropado desde el principio. Trabajar con profesionales entregados es una delicia.

Y hablando de profesionales entregados, quiero hacer una mención a mis compañeros, no solo del TFC, sino de todo el ciclo formativo, Café con Palito, Albano, Daniel y Carmen. Ha sido un placer y un honor trabajar con ellos. Un privilegio haber visto cómo crecían y superaban todos los obstáculos a los que nos enfrentábamos, y de lo más increíble que he experimentado, cómo procuraban que nadie quedase atrás. De las personas más generosas que he encontrado en mi vida. Como siempre, no hay sorpresa, el trabajo en equipo ha sido exquisito. Creo que se ve reflejado en el resultado final.



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

Albano Díez de Paulino:



albano

Este proyecto me ha gustado mucho. Me ha servido para tener un aprendizaje continuo entre la empresa y el proyecto. Además, trabajar en un equipo como Café Con Palito me ha servido para ver que entre varias personas que se entienden y se complementan es posible avanzar y disfrutar del camino.

Por otro lado, me parece que el módulo de proyecto está mal planteado de base. A nadie le apetece, tras 8 horas diarias de trabajo programando, seguir en casa 2 horas más. Además, si has llegado a este punto del curso ya has demostrado que sabes programar, habiendo aprobado todos los módulos. Yo orientaría el módulo exclusivamente al aprendizaje de la gestión integral de un proyecto de software, ya que esto no supone un gran esfuerzo para el alumno y le permite aprender cosas que no ha visto a lo largo de su formación.



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma
Título: Proyecto Chikara

Daniel Espinosa García:



Partiendo de la base que el proyecto es la unión de todo lo aprendido a lo largo de dos años, incluido los conocimientos adquiridos en las FTC, Muestra lo que somos capaces de realizar, tanto de manera individual como colectiva.

En el TFC fuimos capaces de aprender nuevos lenguajes de programación y adaptarnos a nuevas tecnologías, gracias a las bases que adquirimos en este tiempo. Aunque esto supuso un esfuerzo añadido, pudimos superarlo.

Todo lo anterior fue posible gracias a la ayuda de mis compañeros. Los de clase, a lo largo del curso, pero, sobre todo, Carmen, Albano y Ramiro. Los integrantes de Café con Palito, que surgió casi como una broma y se convirtió, a lo largo de estos dos años, en un pilar importante para afrontar retos y aprender cosas nuevas cada día, apoyándonos mutuamente. Esto, sin lugar a dudas, es la ganancia personal más importante que me llevo.

BIBLIOGRAFÍA

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

13. BIBLIOGRAFÍA

Bibliografía

AristiDevs. (2024). *Github*. Obtenido de <https://github.com/ArisGuimera>

AristiDevs. (2024). *Youtube*. Obtenido de <https://www.youtube.com/c/aristidevs>

GitLab B.V. (2024). *GitLab Docs*. Obtenido de <https://docs.gitlab.com/>

JetBrains. (2024). *kotlinlang*. Obtenido de <https://kotlinlang.org/docs/home.html>

Medium. (2024). *Medium*. Obtenido de <https://medium.com/>

Microsoft. (2024). *Microsoft Learn*. Obtenido de <https://learn.microsoft.com/es-es/azure>

OpenAI. (2024). *ChatGPT*. Obtenido de <https://chatgpt.com/>

Stack Exchange Inc. (2024). *Stack Overflow*. Obtenido de <https://stackoverflow.com/>

Stack Exchange Inc. (2024). *Stack Overflow en Español*. Obtenido de <https://es.stackoverflow.com/>

Tiangolo. (2024). *FastAPI Docs*. Obtenido de <https://www.youtube.com/c/aristidevs>

ÍNDICES



14. ÍNDICES

14.1. Índice de Ilustraciones

| | |
|-------------------------------------------------|----|
| CAPTURA 1 – SECURE FILES GITLAB | 42 |
| CAPTURA 2 - VARIABLES DE ENTORNO GITLAB..... | 42 |
| CAPTURA 3 - PIPELINES GITLAB | 43 |
| CAPTURA 4 - TEST UNITARIOS | 75 |
| CAPTURA 5 - TEST INSTRUMENTACIÓN..... | 75 |
| CAPTURA 6 - JMETER Nº MÁXIMO DE PETICIONES..... | 76 |
| CAPTURA 7 - JMETER USUARIOS SIMULTÁNEOS | 77 |
| CAPTURA 8 - AZURE PLAN APP WEB | 77 |
| CAPTURA 9 - REGISTRO DE USUARIOS..... | 78 |
| CAPTURA 10 - TOAST | 79 |
| CAPTURA 11 - GENERICTOAST | 80 |
| CAPTURA 12 - USARNAME ERROR | 80 |
| CAPTURA 13 - PASSWORD ERROR..... | 80 |
| CAPTURA 14 - ANUNCIO IDEALISTA - FACHADA | 98 |
| CAPTURA 15 - ANUNCIO IDEALISTA - PLANO | 98 |
| CAPTURA 16 - GOOGLE MAPS | 99 |

| | |
|----------------------------------------|----|
| CÓDIGO 1 - CHIK JSON | 32 |
| CÓDIGO 2 – ARCHIVO .GITLAB.YML..... | 41 |
| CÓDIGO 3 - NETWORKMODULE | 46 |
| CÓDIGO 4 - REGISTERAPISERVICE | 47 |
| CÓDIGO 5 - REGISTERREPOSITORYIMPL..... | 49 |
| CÓDIGO 6 - CHICKDTO | 50 |
| CÓDIGO 7 - CHICKREPOSITORY..... | 51 |
| CÓDIGO 8 - CHICKUSECASES | 53 |
| CÓDIGO 9 - WELCOMEACTIVITY | 56 |
| CÓDIGO 10 - WELCOMESTATE..... | 57 |
| CÓDIGO 11 - WELCOMEVIEWMODEL | 59 |
| CÓDIGO 12 - USERPREFERENCES | 61 |
| CÓDIGO 13 - ENDPOINT..... | 62 |
| CÓDIGO 14 - SERVICE..... | 63 |
| CÓDIGO 15 - MODELO USER | 64 |



| | |
|---------------------------------------|----|
| CÓDIGO 16 - REPOSITORIO..... | 65 |
| CÓDIGO 17 - MIDDLEWARE JWT | 68 |
| CÓDIGO 18 - DOCKERFILE | 71 |
| CÓDIGO 19 - .ENV | 71 |
| CÓDIGO 20 - VALIDATEFIELDS..... | 79 |
| | |
| DIAGRAMA 1 - GANTT P1 | 23 |
| DIAGRAMA 2 - GANTT P2 | 24 |
| DIAGRAMA 3 - GANTT P3 | 25 |
| DIAGRAMA 4 - MOCKUP | 29 |
| DIAGRAMA 5 - DIAGRAMA DE USO | 30 |
| DIAGRAMA 6 - MODELO RELACIONAL..... | 31 |
| DIAGRAMA 7 - ESTRUCTURA CHIKARA | 33 |
| DIAGRAMA 8 - ESTRUCTURA AZURE | 39 |
| | |
| ILUSTRACIÓN 1 - LOGOTIPO CCP | 88 |

ANEXOS

| | |
|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma |
| | Título: Proyecto Chikara |

ANEXOS

Anexo I Registro Horario

- Solo se muestran los registros iniciales, se entrega un excel con todos

| Issue Key | Issue Summary | Time spent | Date Start | Hour Start | Comment | Author | Epic Key | Epic Summary |
|-----------|------------------------|------------|------------|------------|----------------------------|--------|----------|--------------|
| CHK-3 | REUNIONES INTERNAS | 04:00 | 06/03/2024 | 15:00:00 | KickOff | Albano | CHK-1 | GENERAL |
| CHK-3 | REUNIONES INTERNAS | 04:00 | 06/03/2024 | 15:00:00 | KickOff | Daniel | CHK-1 | GENERAL |
| CHK-3 | REUNIONES INTERNAS | 04:00 | 06/03/2024 | 15:00:00 | kickoff | Ramiro | CHK-1 | GENERAL |
| CHK-3 | REUNIONES INTERNAS | 00:30 | 07/03/2024 | 9:16:51 | Daily | Albano | CHK-1 | GENERAL |
| CHK-9 | DDL Script MySQL | 01:30 | 07/03/2024 | 9:43:03 | | Daniel | CHK-6 | BBDD |
| CHK-9 | DDL Script MySQL | 00:29 | 07/03/2024 | 11:35:35 | | Daniel | CHK-6 | BBDD |
| CHK-9 | DDL Script MySQL | 03:00 | 07/03/2024 | 12:09:01 | | Daniel | CHK-6 | BBDD |
| CHK-9 | DDL Script MySQL | 01:30 | 07/03/2024 | 13:41:06 | Creando scripts con Daniel | Ramiro | CHK-6 | BBDD |
| CHK-9 | DDL Script MySQL | 01:24 | 08/03/2024 | 10:13:58 | | Daniel | CHK-6 | BBDD |
| CHK-10 | DML Script MySQL | 01:00 | 08/03/2024 | 10:54:14 | Añadidos datos de prueba | Daniel | CHK-6 | BBDD |
| CHK-11 | Crear proyecto inicial | 01:29 | 08/03/2024 | 11:39:34 | | Albano | CHK-5 | API REST |

| | |
|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Anexo II Estatutos de Empresa

*Solo se muestra 2 páginas de 17

892.19/001-_____

ESTATUTOS DE "CAFÉ CON PALITO, SOCIEDAD LIMITADA".

TÍTULO I

DENOMINACIÓN, OBJETO, DURACIÓN Y DOMICILIO.

ARTÍCULO 1. Bajo la denominación "CAFÉ CON PALITO, SOCIEDAD LIMITADA", se constituye una Sociedad de Responsabilidad Limitada que se regirá por los presentes Estatutos y, en lo que en ellos no estuviere previsto, por el Real Decreto Legislativo 1/2010, de 2 de julio, por el que se aprueba el texto refundido de la Ley de Sociedades de Capital (en adelante "La Ley" o "Ley de Sociedades de Capital") y demás disposiciones que le sean aplicables.

ARTÍCULO 2. La Sociedad tiene por OBJETO: "El diseño, comercialización, venta y cesión de uso del software. La intermediación y comercialización de servicios relacionados con el software de todo tipo de empresas."

Las actividades integrantes del objeto social podrán ser desarrolladas por la sociedad, total o parcialmente, de modo indirecto, mediante la titularidad de acciones o de participaciones en sociedades con objeto idéntico o análogo.

Quedan excluidos del objeto social todas aquellas actividades para cuyo ejercicio la Ley exija requisitos especiales que no queden cumplidos por esta Sociedad.

ARTÍCULO 3. Si las disposiciones legales exigiesen para el ejercicio de alguna de las actividades comprendidas en el objeto social algún título profesional, autorización administrativa o inscripción en Registros públicos, dichas actividades deberán realizarse por medio de persona que ostente dicha titularidad profesional y, en su caso, no podrán iniciarse antes de que se hayan cumplido los requisitos administrativos exigidos.

ARTÍCULO 4. El DOMICILIO de la Sociedad se establece en el término municipal de El Astillero (Cantabria), sito en Guarnizo, "Polígono Industrial de Guarnizo", barrio La

-Pág 1 -



Estación, número 72-A, oficina 102.

Por acuerdo o decisión del órgano de administración podrá cambiarse el domicilio social dentro del territorio nacional; así como crearse, trasladarse o suprimirse las sucursales, agencias o subdelegaciones, en cualquier lugar del territorio nacional o del extranjero, que el desarrollo de la actividad haga necesario o conveniente.

ARTÍCULO 5. La Sociedad tendrá una DURACIÓN de carácter INDEFINIDO y da comienzo a sus actividades en la fecha de otorgamiento de la escritura de constitución.

TÍTULO II

CAPITAL SOCIAL Y PARTICIPACIONES.

ARTÍCULO 6. El CAPITAL SOCIAL es de TRES MIL EUROS (3.000), dividido en trescientas participaciones sociales, números UNO (1) al TRESCIENTOS (300), ambos inclusive; cada una de ellas tendrá un valor nominal de DIEZ EUROS (10); serán acumulables e indivisibles, y no podrán incorporarse a títulos negociables ni denominarse acciones.

El capital social está íntegramente suscrito y desembolsado.

Las participaciones atribuyen a los socios los mismos derechos.

ARTÍCULO 7. Las participaciones sociales no se representarán en ningún caso por títulos, nominativos o al portador, ni mediante anotaciones en cuenta; tampoco se expedirán resguardos provisionales acreditativos de una o varias participaciones sociales. El único título de propiedad estará constituido por la escritura pública de constitución o de aumento de capital o por los documentos públicos que, según los casos, acrediten las transmisiones posteriores.

Las certificaciones del libro registro de socios en ningún caso sustituirán al título público de adquisición.

ARTÍCULO 8.

1.- Transmisión voluntaria: Es libre la transmisión voluntaria de participaciones sociales por actos inter vivos cuando tenga lugar entre socios. También serán libres las transmisiones realizadas por un socio a favor de su cónyuge, ascendiente o descendiente o, en su caso, la realizada a favor de sociedades pertenecientes al mismo grupo



Anexo III Modelo 036



Agencia Tributaria

Teléfono: 901 33 55 33
www.agenciatributaria.es

ANEXO II

DECLARACIÓN CENSAL

de alta, modificación y baja en el Censo de Empresarios, Profesionales y Retenedores

Pág. 1
**Modelo
036**

Datos identificativos

Espacio reservado para la etiqueta identificativa.

| | | | | | |
|----------------------|-----|-----|-----------------------------------------|-----|--------|
| 101 | NIF | 102 | Apellidos o razón o denominación social | 103 | Nombre |
| Café Con Palito S.L. | | | | | |

Espacio reservado para numeración por código de barras

1. CAUSAS DE PRESENTACIÓN

A) Alta

- 110 Solicitud de Número de Identificación Fiscal (NIF)
 111 Alta en el censo de empresarios, profesionales y retenedores

B) Modificación

- 120 Solicitud de NIF definitivo, disponiendo de NIF provisional.
 121 Solicitud de nueva tarjeta acreditativa del NIF
 142 Modificación de datos de teléfonos y direcciones electrónicas. (páginas 2A, 2B y 2C)
 122 Modificación domicilio fiscal. (páginas 2A, 2B y 2C)
 123 Modificación domicilio social o de gestión administrativa. (páginas 2A y 2B)
 124 Modificación y baja domicilio a efectos de notificaciones. (páginas 2A, 2B y 2C)
 125 Modificación otros datos identificativos. (páginas 2A, 2B y 2C)
 126 Modificación datos representantes. (página 3)
 127 Modificación datos relativos a actividades económicas y locales. (página 4)
 128 Modificación de la condición de Gran Empresa o Admón. Pública de presupuesto superior a 6.000.000 de euros. (página 5)
 129 Solicitud de alta/baja en el registro de devolución mensual. (página 5)
 130 Solicitud de alta/baja en el registro de operadores intracomunitarios. (página 5)
 143 Comunicación de opción y renuncia a la llevanza de los Libros registro del IVA a través de la Sede electrónica de la AEAT. (página 5)
 131 Modificación datos relativos al Impuesto sobre el Valor Añadido. (página 5)
 132 Modificación datos relativos al Impuesto sobre la Renta de las Personas Físicas. (página 6)
 133 Modificación datos relativos al Impuesto sobre Sociedades. (página 6)
 134 Modificación datos relativos al Impuesto sobre la Renta de no Residentes correspondiente a establecimientos permanentes o a entidades en atribución de rentas constituidas en el extranjero con presencia en territorio español. (página 6)
 135 Opción/renuncia por el Régimen fiscal especial del Título II de la Ley 49/2002. (página 6)
 136 Modificación datos relativos a retenciones e ingresos a cuenta. (página 7)
 137 Modificación datos relativos a otros Impuestos. (página 7)
 138 Modificación datos relativos a regímenes especiales del comercio intracomunitario. (página 7)
 139 Modificación datos relativos a la relación de socios, miembros o participes. (página 8)
 140 Dejar de ejercer todas las actividades empresariales y/o profesionales (personas jurídicas y entidades, sin liquidación. Entidades inactivas).

Fecha efectiva del cese
 141 / /

C) Baja

- 150 Baja en el censo de empresarios, profesionales y retenedores.

151 Causa

Fecha efectiva de la baja
 152 / /

Lugar, fecha y firma

Lugar

Santander, Cantabria

Fecha

19 de Diciembre de 2023

Firma en calidad de

Firma

Firmado D./D^a:

Ejemplar para el interesado



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma

Título: Proyecto Chikara

**Modelo
036**

NIF

Apellidos y nombre o razón o denominación social

Pág. 2A

Espacio reservado para N° justificante

Hoja .../...

2. IDENTIFICACIÓN

A) Personas físicas

| | | | | |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|---------------------------------|----------------------------|---------------------------------------------------|
| A1 Residente fiscal en España <input checked="" type="checkbox"/> | A2 No residente fiscal en España <input type="checkbox"/> | A3 Nacionalidad Española | | |
| A26 Diplomático español o asimilado (art. 10 LIRPF) <input type="checkbox"/> | A27 Diplomático extranjero en España o asimilado (art. 9.2 LIRPF) <input type="checkbox"/> | | | |
| Identificación | | | | |
| A4 NIF 72174805 T | A5 Apellido 1 Espinosa | A6 Apellido 2 García | A7 Nombre Daniel | A8 Nombre comercial Cafe Con Palito S.L |

A90 Código identificación fiscal del Estado de residencia/NIF-IVA (NVAT)

A9 Condición de "Emprendedor de responsabilidad limitada"

Alta Baja

19 / 12 / 2023

A10 Fecha de inscripción o cancelación como emprendedor de responsabilidad limitada en el Registro Mercantil

| | | | |
|---------------------------------------------------------------|----------------------------------------------|-------------------------------------|----------------------------|
| A28 Tfno. Fijo nacional 696803992 | A29 Tfno. Móvil nacional 696803992 | A38 Tfno. Fijo extranjero | A39 Tfno. Móvil extranjero |
| A40 Correo electrónico despinosag02@educantabria.es | | A56 Dominio o dirección de Internet | |

Domicilio fiscal en España

| | | | | | | | | | |
|-------------------------------------------------------------------------------------|------------------------------|---------------------------|---------------|---------------|-----------------------------------------------------------------------------|------------|------------|------------|------------|
| A11 Tipo de vía Calle Primero de Mayo | A12 Nombre de la vía pública | A13 Tipo Num. 1 | A14 Núm. casa | A15 Calif. nu | A16 Bloque | A17 Portal | A18 Escal. | A19 Planta | A20 Puerta |
| A21 Complemento domicilio (ej: Urbanización., Polígono Industrial., C. Comercial..) | | | | | A22 Localidad / Población (si es distinta de Municipio) Santander | | | | |
| A24 Nombre del Municipio | | | | | A25 Provincia Cantabria | | | | |

Domicilio fiscal en el estado de residencia (no residentes)

| | | |
|------------------------------------------------|------------------------|---------------------|
| A31 Domicilio (Address) | A34 Población / Ciudad | A33 C. Postal (ZIP) |
| A32 Complemento domicilio (si fuese necesario) | A36 País | A37 Cod. País |
| A35 Provincia / Región / Estado | | |

Domicilio a efectos de notificaciones (si es distinto del fiscal, cumplimente el apartado 1 ó el 2 según estime oportuno)

| | | | | | | | | | | |
|-------------------------------------------------------------------------------------|-----------------|------------------------------|---------------|---------------|-----------------------------------------------------------------|------------|------------|------------|------------|-----------------------------------------------------------------|
| A40 Baja | A41 Tipo de vía | A42 Nombre de la vía pública | A43 Tipo Num. | A44 Núm. casa | A45 Calif. nu | A46 Bloque | A47 Portal | A48 Escal. | A49 Planta | A50 Puerta |
| A51 Complemento domicilio (ej: Urbanización., Polígono Industrial., C. Comercial..) | | | | | A52 Localidad / Población (si es distinta de Municipio) | | | | | A53 C. Postal |
| A54 Nombre del Municipio | | | | | A55 Provincia | | | | | |
| A59 Destinatario (si es distinto del declarante) | | | | | A60 En calidad de: (representante, apoderado, familiar, etc...) | | | | | |
| 2) A61 APARTADO DE CORREOS NÚMERO: | | | | | A62 Población / Ciudad | | | | | A63 C. Postal |
| A64 Provincia | | | | | A68 Destinatario (si es distinto del declarante) | | | | | A69 En calidad de: (representante, apoderado, familiar, etc...) |

Domicilio gestión administrativa

| | | | | | | | | | | |
|-------------------------------------------------------------------------------------|------------------------------|---------------|---------------|---------------|---------------------------------------------------------|------------|------------|------------|------------|---------------|
| A71 Tipo de vía | A72 Nombre de la vía pública | A73 Tipo Num. | A74 Núm. casa | A75 Calif. nu | A76 Bloque | A77 Portal | A78 Escal. | A79 Planta | A80 Puerta | |
| A81 Complemento domicilio (ej: Urbanización., Polígono Industrial., C. Comercial..) | | | | | A82 Localidad / Población (si es distinta de Municipio) | | | | | |
| A83 C. Postal | | | | | A84 Nombre del Municipio | | | | | A85 Provincia |

Establecimientos permanentes

| | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-----------------------------|----------------------------------------|
| A91 ¿Opera en España a través de establecimiento permanente? | Sí <input type="checkbox"/> | NO <input type="checkbox"/> | A92 ¿Cuántos? <input type="checkbox"/> |
| Identifique los establecimientos permanentes a través de los que opera en España. Indique sólo la denominación diferenciada asignada a cada uno: | | | |

| | |
|--------------------|--|
| A94 1 Denominación | |
| A96 2 Denominación | |
| A98 3 Denominación | |

Ejemplar para el interesado

Díez de Paulino, Albano
Espinosa García, Daniel
Gutiérrez Valverde, Ramiro

[123]

PREGUNTAS FRECUENTES

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  INFORMATICA IES AGI | Ciclo: Desarrollo de Aplicaciones Multiplataforma Modulo: Proyecto De Desarrollo De Aplicaciones Multiplataforma Título: Proyecto Chikara |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

PREGUNTAS FRECUENTES

1. ¿Por qué se ha usado un modelo de datos relacional y un modelo de datos no-relacional conjuntamente?

Aunque hay maneras de utilizar solamente un tipo de modelo, que hemos descartado, optamos por utilizar ambos porque así podemos ofrecer una mejor gestión de los datos.

En el modelo relacional almacenamos los datos del usuario, que son estables y fijos, para obtener un mayor rendimiento, consistencia y fiabilidad en las consultas. Sin embargo, la gestión de los chiks es diferente porque cada uno de ellos tiene unas determinadas características que lo hacen diferente del resto como, por ejemplo, la cantidad de imágenes, textos o videos y su orden. Por eso necesitamos utilizar un modelo más flexible y eficiente para esta casuística.

2. ¿Cómo se relacionan los modelos de datos usados?

Los modelos se relacionan a través de la ID del usuario. Tanto PostgreSQL en cada registro, como MongoDB en cada uno de los chiks insertados almacenan la información de la ID del usuario.

3. ¿Por qué hemos usado el servicio cloud Azure de Microsoft sobre otras alternativas?

En Café con Palito tenemos una predilección por el sistema de microservicios de Microsoft, que encontramos más práctico que el de AWS o Google, por poner otros ejemplos. Pero el verdadero motivo por el que nos decantamos por el servicio cloud de Azure es que nos ofrece una redundancia geográfica controlada. Los datos de los usuarios de la aplicación quedarán registrados en países dentro del Espacio Económico Europeo (EEE), o en aquellos que cumplan con el Reglamento General de Protección de Datos (RGPD), siendo posible configurar la elección en todo momento.



4. ¿Por qué usamos una arquitectura de microservicios en vez de una monolítica?

Nuestro principal motivo para utilizar la arquitectura basada en microservicios es su escalabilidad. Escalabilidad a la hora de implementar servicios de forma independiente, pero también horizontalmente, permitiendo manejar grandes volúmenes de tráfico simultáneamente, bajo demanda.

5. ¿El desarrollo móvil es solo para dispositivos con SO Android, tenemos planes para un desarrollo para SO iOS? ¿Qué lenguaje usaríamos?

Chikara nace con un corazón multiplataforma. Que estemos presentando una aplicación móvil se debe a que nos parece el sitio lógico por el que empezar, dado que, según entendemos, es la manera correcta de consumir los chiks, por la absoluta disponibilidad que ofrecen este tipo de dispositivos. IOS, Xcode y su lenguaje más utilizado Swift, son la siguiente parada lógica del proyecto. Si todo evoluciona como hemos previsto y cumplimos con los objetivos de acogida que consideramos aceptables, no tardará en estar disponible una versión para IOS de Chikara con las mismas prestaciones que la de Android.

6. ¿Si se deja planteado un desarrollo para múltiples dispositivos porque no hemos usado un framework multiplataforma?

Para responder a esta pregunta necesitaríamos abrir el debate de si es mejor utilizar un framework nativo, o si las ventajas que este ofrece no son suficientes para desechar uno multiplataforma. No es el sitio adecuado para ello. Hemos escogido funcionalidad sobre facilidad.