

**DESARROLLO DE APLICACIÓN MÓVIL ANDROID CON SERVICIOS MEDIANTE
SPRING BOOT PARA APOYO AL APRENDIZAJE DEL IDIOMA FRANCÉS POR
MEDIO DE LA GAMIFICACIÓN**

Autor:

STEFANY JULIET FLÓREZ HERNÁNDEZ



Directora: M.sc. Rocío De Belén Contreras Manrique

**Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Departamento de Ingenierías Electrónica, Eléctrica, Sistemas y Telecomunicaciones, EEST
Ingeniería de Sistemas
Villa del Rosario, Colombia
2022**

Dedicatoria

Dedico este proyecto a mis padres y hermanos, quienes han estado conmigo desde el comienzo, en cada etapa de mi proceso de formación, me han ayudado a salir adelante, y me han visto llegar hasta acá, y para quienes espero ser una inspiración, así como ellos lo han sido para mí.

Agradecimientos

Agradezco a mi familia por su apoyo y motivación en todo momento durante este proceso.

A mis profesores y todas las personas que han contribuido en mi educación.

A mi tutora Rocío de Belén Contreras Manrique, por su importante colaboración.

Resumen

En el presente documento se expone el desarrollo de un sistema compuesto por una aplicación móvil, una plataforma web y backend de servicios REST, realizado para contribuir al campo de la enseñanza de idiomas, específicamente el francés, aprovechando las ventajas del m-learning y la gamificación, para solucionar inconvenientes propios del aprendizaje tradicional como la necesidad de estudiar en lugares y horarios específicos o la falta de motivación al estudiar mediante lectura de libros o clases en aulas.

El objetivo de este proyecto es desarrollar una aplicación que sirva de apoyo al proceso de enseñanza-aprendizaje del idioma francés, que permita adquirir conocimiento y practicarlo de manera didáctica, presentando la información en forma llamativa mediante la utilización de colores, imágenes y audios, haciendo uso, también, de gamificación para la sección de práctica.

Para llevar a cabo el proyecto, se empleó la metodología Mobile-d, la cual se caracteriza por generar un producto final en un tiempo corto; se utilizaron, además, herramientas como Spring Boot para la creación de los servicios, el framework Angular para el desarrollo de la plataforma web, Android Studio para la aplicación móvil y los lenguajes de programación Java y JavaScript.

Palabras claves: Aplicación móvil, Android, gamificación, francés, mobile-d, m-learning.

Abstract

This document presents the development of a system made up of a mobile application, a web platform and a backend of REST services, made to contribute to the field of language teaching, specifically French, taking advantage of m-learning and gamification, to solve problems typical of traditional learning such as the need to study in specific places and times or the lack of motivation when studying by reading books or taking classes in classrooms.

The objective of this project is to develop an application that supports the teaching-learning process of the French language, allowing knowledge to be acquired and practiced in a didactic way, attractively presenting information through the use of colors, images and audio, making use, also, of gamification for the practice section.

To carry out the project, the Mobile-d methodology was used, which is characterized by generating a final product in a short time; in addition, tools were used, such as Spring Boot to create the services, the Angular framework for the development of the web platform, Android Studio for the mobile application, and the Java and JavaScript programming languages.

Keywords: Mobile application, Android, gamification, French, mobile-d.

Tabla de Contenido

Introducción	17
Capítulo 1. Generalidades	18
Planteamiento del Problema	18
Formulación del Problema	19
Justificación	19
Objetivos	20
Objetivo general	20
Objetivos específicos	20
Acotaciones	20
Capítulo 2. Marco Teórico	22
Antecedentes	22
Antecedentes internacionales	22
Antecedentes nacionales	23
Antecedentes regionales	24
Marco Teórico	24
Aprendizaje de idiomas	24
Estilos de Aprendizaje	25
Aprendizaje en la app Voilá.	26
Gamificación	26
Elementos del juego.	26
Metodologías de desarrollo ágil para aplicaciones móviles	27
Metodología mobile-D.	27
Hybrid methodology design.	28

Mobile development process Spiral. -----	28
Herramientas Tecnológicas-----	28
Android Studio. -----	28
Spring Boot. -----	29
Angular. -----	30
Microservicios. -----	31
Pruebas de Software. -----	32
Marco Legal.....	32
ISO 21001-----	32
Ley 1341 del 30 de julio de 2009. Artículo 39 -----	32
Ley 1955 de 2019 -----	32
Decreto 393 de 1991 -----	33
Capítulo 3. Metodología de Desarrollo Ágil-----	34
Fase de Exploración.....	34
Inicialización.....	35
Estabilización.....	35
Pruebas del Sistema	35
Capítulo 4. Desarrollo del Sistema -----	36
Fase de Exploración.....	36
Requisitos Comunes de los Interfaces -----	36
Interfaces de usuario. -----	36
Requisitos de Hardware. -----	36
Requisitos de software.-----	36

Requisitos Funcionales-----	37
Requisito funcional 1 -----	37
Requisito funcional 2 -----	37
Requisito funcional 3 -----	38
Requisito funcional 4 -----	38
Requisito funcional 5 -----	38
Requisito funcional 6 -----	39
Requisito funcional 7 -----	39
Requisito funcional 8 -----	40
Acotaciones -----	40
Diagrama de Casos de Uso -----	40
Comparación de herramientas TIC -----	41
Fase de Inicialización.....	43
Contenido temático -----	43
Esquema de arquitectura -----	48
Diseño de interfaces -----	49
Diseño de base de datos -----	52
Fase de Producción	53
Desarrollo de aplicación móvil -----	53
Arquitectura MVP.-----	53
Interfaces gráficas.-----	54
Desarrollo de las lecciones.-----	61
Pronunciación.-----	64

Gamificación.-----	64
Pruebas unitarias y de integración. -----	73
Desarrollo de plataforma administrativa -----	76
Arquitectura modular. -----	76
Componentes. -----	76
Desarrollo de backend-----	77
Arquitectura de microservicios.-----	77
Arquitectura MVC. -----	80
Seguridad en el registro de usuarios.-----	81
Base de datos -----	82
Fase de Estabilización.....	83
Integración de la aplicación móvil con el backend -----	83
Registro de usuarios. -----	84
Envío de token para inicio de sesión. -----	86
Integración de plataforma web con el backend-----	88
Listar usuarios con avance. -----	89
Fase de Pruebas del Sistema	90
Pruebas de rendimiento -----	90
Pruebas de carga -----	95
Conclusiones -----	96
Referencias -----	97

Lista de Figuras

Figura 1	logo del IDE Android Studio	29
Figura 2	Logo de Spring.....	29
Figura 3	Logo de Angular	30
Figura 4	Arquitectura basada en microservicios.....	31
Figura 5	Ciclo de desarrollo de la metodología Mobile-D.....	34
Figura 6	Diagrama de casos de uso	41
Figura 7	Esquema de arquitectura voilá	49
Figura 8	Bosquejo de pantalla de inicio de sesión.....	50
Figura 9	Bosquejo de pantalla de registro de usuario	50
Figura 10	Bosquejo de pantalla de lección	51
Figura 11	Bosquejo de Menu de lecciones	51
Figura 12	Modelo relacional base de datos remota	52
Figura 13	Modelo relacional base de datos local	53
Figura 14	Paquete MVP en Android Studio.....	54
Figura 15	Paquete de layout en Android Studio	55
Figura 16	Codigo XML y vista previa de interfaz gráfica	56
Figura 17	Clases java pertenecientes a la capa de vista	57
Figura 18	Icono de app Voilá en el dispositivo móvil	58
Figura 20	Cuadro de diálogo con animación para juego ganado	59
Figura 21	Cuadro de diálogo con animación para juego perdido	60
Figura 22	Lección Saludos	61

Figura 23 Lección Números	62
Figura 24 Diseño tarjeta para mostrar contenido de lecciones	63
Figura 25 Constructor de clase adaptador	63
Figura 26 Inicialización de TexToSpeech	64
Figura 27 Diseño de juego ahorcado	65
Figura 28 Juego ahorcado en lección saludos.....	66
Figura 29 Diseño de juego memoria.....	67
Figura 30 Juego memoria en lección números	68
Figura 31 Diseño juego ruleta.....	69
Figura 32 Juego ruleta en lección colores	70
Figura 33 Diseño de juego de arrastrar	71
Figura 34 Juego arrastrar en lección estacioens	72
Figura 35 Prueba unitaria para validar correo	73
Figura 36 Resultado de prueba unitaria para validar correo.....	74
Figura 37 Prueba de integración para creación de usuario	75
Figura 38 Resultado de prueba de integración para creación de usuario	75
Figura 39 Módulo Core – Feature – Shared en angular	76
Figura 40 Componente listar- usuario	77
Figura 41 Componente para usuarios.....	78
Figura 42 Componente para lecciones	78
Figura 43 Conexión a base de datos y configuración de puerto de componente	79
Figura 44 Conexión a base de datos y configuración de puerto de componente	79
Figura 45 Arquitectura MVC en componente usuarios	80

Figura 46 Metodo de registro de usuario usando encriptación de contraseña	81
Figura 47 Usuario registrado en base de datos con contraseña encriptada	81
Figura 48 Validación de contraseña para inicio de sesión	82
Figura 49 Tablas de la base de datos en MySQL.....	82
Figura 50 Disparador para creación de lecciones.....	83
Figura 51 Formulario para registro de usuario	84
Figura 52 Registro de usuario empleando librería Retrofit	85
Figura 53 Declaración de método rgitrar usuario en API	85
Figura 54 Obtención de respuesta en Retrofit	86
Figura 55 Método para generar token en Android Studio	87
Figura 56 Consumo de servicio para incio de sesión enviado token generado	87
Figura 57 Validaciones en Spring Boot para incio de sesión	88
Figura 58 Método para verificación de token en Spring Boot.....	88
Figura 59 Vista de la Plataforma con los resultado de usuarios con avance.....	89
Figura 60 Consumo de servicio para la consulta de usuarios con avance	89
Figura 61 Asignación de los datos obtenidos en el consumo del servicio	90
Figura 62 Estructura HTML para mostrar los datos obtenidos	90
Figura 63 Medición de uso de recursos en dispositivo Xiaomi durante el incio	91
Figura 64 Uso de CPU al entrar a la vista de inicio de sesión.....	92
Figura 65 Uso de memoria al entrar a la vista de inicio de sesión.....	92
Figura 66 Medición de uso de CPU al presionar el botón INICIAR SESIÓN.....	93
Figura 67 Medición del uso de memoria al presionar el botón INICIAR SESIÓN.....	94
Figura 68 Medición del uso de red al presionar el botón INICIAR SESIÓN	94

Figura 69 Configuración de número de peticiones y tiempo	95
Figura 70 Resultados de la prueba de carga para consulta de usuario	95

Lista de Tablas

Tabla 1. Requisito funcional 1	37
Tabla 2 Requisito Funcional 2	37
Tabla 3 Requisito Funcional 3	38
Tabla 4 Requisito Funcional 4	38
Tabla 5 Requisito Funcional 5	38
Tabla 6 Requisito Funcional 6	39
Tabla 7 Requisito Funcional 7	39
Tabla 8 Requisito Funcional 8	40
Tabla 9 Comparación de herramientas de tecnologías de la información	42

Introducción

Actualmente los dispositivos móviles son elementos esenciales para nuestra vida diaria, por lo que las aplicaciones para estos dispositivos, se han convertido en herramientas de gran importancia en la mayoría de áreas; hoy en día se pueden encontrar aplicaciones para campos como la salud, deportes, entretenimiento, o educación, entre muchos otros, siendo este último el de mayor interés en el presente proyecto.

Ha sido tal la trascendencia de los dispositivos móviles en el área de la educación, que hoy en día existe un concepto de gran relevancia conocido como m-learning, que consiste básicamente en el aprendizaje que se realiza a través de un dispositivo móvil, este tipo de aprendizaje ha sido ampliamente estudiado y acogido debido a las ventajas que presentan frente a otros métodos de aprendizaje, la mayoría de estas ventajas basadas en la movilidad, ya que como lo afirman Kumar et al. (2018) con m-learning, el aprendizaje se puede dar en cualquier momento y cualquier lugar.

Lo que se busca con este proyecto es aprovechar las ventajas del m-learning, al desarrollar una aplicación educativa, incorporando además elementos de juego, al emplear gamificación como estrategia didáctica en el proceso de enseñanza-aprendizaje del idioma francés, para que, de esta forma, los usuarios puedan lograr una mejor retención de los conocimientos adquiridos con el uso de la aplicación, al aumentar el tiempo de práctica gracias a la motivación que el juego puede generar, pues como considera Kapp (2012), los juegos atraen a los estudiantes por el placer de jugar y tener una experiencia atractiva de aprendizaje.

Capítulo 1. Generalidades

Planteamiento del Problema

La educación en cualquier área siempre ha sido un tema de alta complejidad debido a su amplitud; existen multitud de estudios sobre la pedagogía y la implementación de diversas estrategias que contribuyan a mejorar el proceso de aprendizaje, y que permitan adaptarse a los desafíos que la educación presenta; uno de los principales retos es que, como lo expone Gualancañay (2018), cada estudiante tiene su propio estilo de aprendizaje, lo que hace que no siempre se adapte al estilo de aprendizaje preferido por su docente, esto lleva a que se requiera un refuerzo por parte del estudiante.

Cuando se trata de aprender idiomas lo que se observa es que, muchas personas recurren a buscar alternativas diferentes al aula, ya sea para reforzar o para, desde un inicio, aprender por su cuenta, es por eso que actualmente se pueden encontrar por ejemplo, multitud de videos sobre idiomas en plataformas como YouTube, pero esto supone otros inconvenientes como el tiempo adicional que se le debe dedicar al refuerzo de lo visto en clase, o si se realiza un aprendizaje del todo autodidacta, la disciplina que se debe tener para aprender por sí mismo, y es en este momento cuando, para ambos casos, el uso de diversas herramientas tecnológicas, entra a jugar un rol esencial.

Como lo relata Guy (2009), el uso de dispositivos móviles para el aprendizaje, nace con la migración de estos dispositivos de sistemas 2G a 3G, ya que esto permitía acceder a internet por medio de Wi-Fi; en junio de 1985, existían 200.000 suscriptores a dispositivos inalámbricos en los Estados Unidos, y diez años después, esta cantidad incrementó a 28 millones, en el 2004, la mayoría de los estudiantes universitarios de Japón llevaban teléfonos móviles y muchos de estos estudiantes leían sus correos electrónicos o ingresaban información en sus teléfonos móviles; para noviembre del 2007, el número de dispositivos móviles alcanzaba 3.3 billones alrededor del mundo. Con el establecimiento de la teoría pedagógica de aprendizaje móvil, muchos países especialmente en

Europa introdujeron exitosamente el aprendizaje móvil en programas de educación a distancia en universidades abiertas.

Formulación del Problema

¿Cómo se puede contribuir, desde el desarrollo móvil, en el aprendizaje del idioma francés mediante el juego?

Justificación

El uso de herramientas tecnológicas para el aprendizaje, en particular los teléfonos inteligentes, presentan importantes ventajas, algunas de estas, planteadas por la UNESCO (2013), son la facilidad para el aprendizaje personalizado y la posibilidad de aprendizaje en cualquier momento y lugar, lo que supone una buena opción para reforzar el aprendizaje del aula aprovechando momentos del día que, mediante otro tipo de herramientas no sería posible, por ejemplo, al desplazarse en transporte público, además de las ventajas de un aprendizaje personalizado como la posibilidad de repetir lecciones las veces que sea necesario o estudiar durante los periodos de tiempo que el estudiante considere adecuados.

El desarrollo de una aplicación para el aprendizaje del francés pretende presentar una alternativa atractiva para los estudiantes, que les permita beneficiarse de las ventajas mencionadas anteriormente, y contribuir al campo del aprendizaje de lenguas extranjeras enfocándose en un idioma que, a pesar de ser la lengua materna de millones de personas en el mundo, y ser un idioma oficial en varias regiones del continente americano, no existen muchas opciones para estudiarlo en Colombia, ya que la mayoría de instituciones se centran, por justas razones, en la enseñanza del inglés.

Además de las ventajas propias de las aplicaciones móviles, este proyecto brinda beneficios importantes en el área del aprendizaje, gracias a la implementación de gamificación, ya que, como mencionan Ortiz et al. (2018), el uso de esta estrategia proporciona incremento de motivación, inmersión y compromiso por parte de los estudiantes, ya que les brinda un aprendizaje más estimulante que mediante otros métodos de enseñanza-aprendizaje.

Objetivos

Objetivo general

Desarrollar la aplicación móvil Android con servicios mediante Spring Boot para apoyo al aprendizaje del idioma francés por medio de la gamificación.

Objetivos específicos

- Comparar las herramientas de tecnologías de la información y comunicaciones para apoyo al aprendizaje en el idioma francés.
- Diseñar la aplicación incluyendo los requerimientos funcionales y no funcionales.
- Realizar la gamificación para la sección de práctica de las lecciones de la aplicación móvil.
- Implementar servicios mediante la herramienta Spring Boot.

Acotaciones

- La aplicación se desarrollará únicamente para dispositivos con sistema Android, versión mínima 4.1 (Jelly Bean).
- Se abarcarán temáticas de francés correspondiente exclusivamente al nivel A1 según el Marco Común Europeo de Referencia para las Lenguas (MCERL).
- La metodología de desarrollo a utilizar para la aplicación móvil es Mobile-D.
- Para el desarrollo del Backend se usará la herramienta Spring Boot.

- El Frontend se compone de dos partes correspondientes a la aplicación móvil, para la cual no se empleará un Framework y una plataforma administrativa en la cual se usará Angular.

Capítulo 2. Marco Teórico

Antecedentes

Antecedentes internacionales

Alrededor del mundo se han desarrollado distintos tipos de aplicaciones y herramientas tecnológicas enfocadas en el aprendizaje de idiomas, principalmente del idioma inglés, tal es el caso de Wu (2015) quién desarrolló una aplicación para Smartphones en China e investigó la efectividad de esta herramienta como ayuda en el aprendizaje de vocabulario en inglés para estudiantes universitarios, comparando los resultados del proceso de aprendizaje de dos grupos de estudiantes, uno sin la aplicación y otro usando la aplicación, y demostrando que su uso mejoraba significativamente el aprendizaje de vocabulario.

En España, Moreno & Vermeulen (2015), desarrollaron una aplicación móvil llamada VISP (Videos for SPEaking), que utiliza técnicas de descripción de audio para promover fluidez y precisión en la producción oral, y presentaron un estudio con los resultados del uso de su aplicación en estudiantes españoles que aprendían inglés como lengua extranjera.

Respecto a la implementación de arquitectura basada en microservicios, cabe mencionar el estudio realizado por Lopez & Maya, (2017) en el que se lleva a cabo un análisis de las tecnologías y arquitectura usada en la Coordinación General de Tecnologías de la información y Comunicación de la Asamblea Nacional de Ecuador, las ventajas que representa el uso de arquitectura basada en microservicios, y la necesidad del cambio de la arquitectura que está siendo empleada.

En cuanto al uso del Framework Angular, se ha llevado a cabo estudios como el de Sahani & Singh, (2020) donde se menciona la importancia del uso de este tipo de herramientas debido a la creciente complejidad en el campo del desarrollo web.

En Brasil, otro tipo de aplicación educativa fue desarrollada por Pontes et al., (2020), en este caso consiste en un juego multiplataforma para la enseñanza de los números en lenguaje de señas brasileño, es basado en turnos, altamente visual y táctil, y está orientado a niños y adultos.

Por otro lado, también en España, Pegalajar (2021), realizó un estudio sobre las implicaciones de la gamificación en la educación superior, en el que destaca potencialidades de esta como el aumento de la motivación, el interés y la participación de los estudiantes en el proceso de enseñanza – aprendizaje, además de una mejora en el rendimiento académico y desarrollo de habilidades y competencias profesionales.

Antecedentes nacionales

En el ámbito nacional, se han desarrollado estudios enfocados en esta área, el trabajo realizado por Gómez & Chacón, (2017), en el que se propone la implementación del aprendizaje móvil en el área de fisioterapia, utilizando la metodología pedagógica del aprendizaje basado en problemas, y se realiza un estudio previo de aplicaciones móviles en el campo de la fisioterapia.

Una tesis de grado desarrollada en relación a este tema, corresponde al nivel de maestría, realizada por Arango Giraldo (2018) y está enfocada en el área de gestión, en este se presentan los resultados de la investigación realizada para conocer la viabilidad de desarrollar y comercializar, en el mercado colombiano, una aplicación digital, que permita la enseñanza del idioma inglés a niños con dislexia, bajo los lineamientos de la metodología ONUDI.

Del mismo modo, el trabajo realizado Sanchez Acosta (2019), que consiste en una aplicación de realidad aumentada para dispositivos Android, para el aprendizaje de inglés nivel A1 en niños de 10 a 12 años de edad, esta aplicación usa reconocimiento basado en marcadores para mostrar en la pantalla del dispositivo una imagen en 3D.

Antecedentes regionales

En el departamento Norte de Santander, se han llevado a cabo proyectos con aplicaciones móviles en variedad de áreas, como el realizado por Álvarez (2018), que busca determinar la pertinencia de usar la metodología ágil scrum en la elaboración de software móvil, para esto, se describen los hitos, herramientas, métodos, roles y artefactos que brinda SCRUM, se describe la conveniencia de utilizarlos en conjunto con las tecnologías disponibles para diseñar para la plataforma Android, y finalmente se implementa esta metodología en la producción sobre requisitos cambiantes de una aplicación.

También se puede encontrar el trabajo realizado por Trillos et al. (2019), quienes desarrollaron una aplicación para la gestión de un proyecto caprino en la Universidad Francisco de Paula Santander sede Ocaña, la aplicación permite la optimización de los procesos y en el artículo publicado se sostiene que, mediante el uso de la esta se logró una rapidez notable y un mayor control en dichos procesos.

En la universidad de Pamplona, Figueroa (2019) llevó a cabo una tesis, que consiste en una aplicación móvil para estudiantes con discapacidad diferencial auditiva, y según argumenta la autora, al implementar la aplicación los estudiantes obtuvieron una mejor comprensión de la clase cuando el intérprete de señas no estaba presente.

Marco Teórico

Aprendizaje de idiomas

Stephen Krashen, perito en lingüística, ha sido reconocido por sus publicaciones sobre la adquisición de una segunda lengua y la importancia de las mismas al tener una influencia en las demás áreas de aprendizaje. La teoría de Krashen de adquisición de segundas lenguas expuesta en el libro "Second Language Acquisition and Second Language Learning" (Krashen, 1981), plantea que,

en un principio, el aprendizaje se da cuando una persona lee o escucha algo sin necesidad de hablarlo o escribirlo en primera estancia, tal como ocurre al ver películas en otro idioma con subtítulos en ese idioma, o en este caso, al usar una aplicación donde se leen palabras, se escucha su pronunciación y algunas veces, se visualiza a qué hace referencia dicho vocabulario.

Krashen (1985), en su teoría, deriva hipótesis de la misma en “The Input Hypothesis. Issues and Implications explicando la diferencia entre la adquisición y el aprendizaje, mostrando el primero como algo inconsciente, como cuando se aprende la lengua nativa, y el segundo como consiente, cuando se aprende la estructura de dicha lengua. Con esta aplicación, se fomentan ambos; es adquisitivo cuando solo se escucha y se relaciona, y de aprendizaje, cuando se habla y se aprenden las reglas de la lengua.

Estilos de Aprendizaje

De acuerdo con el modelo de programación neurolingüística de Grinder & Bandler (2007), en el aprendizaje existen tres sistemas de representación mental de la información: el visual, que permite recordar mediante imágenes, el auditivo que corresponde a los sonidos y el kinestésico que hace referencia a las sensaciones (cutánea, muscular, visceral, emocional). Estos estilos de aprendizaje han sido estudiados y aplicados ampliamente en distintos campos, y el aprendizaje de idiomas no es la excepción, tal es el caso del estudio realizado por Quevedo et al. (2021) en el que emplearon un método auditivo visual para la enseñanza del inglés en estudiantes de turismo, y los resultados demostraron que el uso de dicho método fortaleció el aprendizaje del idioma en los estudiantes.

Aprendizaje en la app Voilá. La aplicación Voilá ha sido diseñada tomando en cuenta los tres sistemas de aprendizaje planteados en el modelo de programación neurolingüística de Grinder & Bandler (2007). En la aplicación se hace uso del sistema visual al presentar información como texto en francés y español acompañado de imágenes o colores representativos según la necesidad de la lección que se esté estudiando; por otro lado, se emplea un sistema auditivo ya que cada lección ofrece audios con la pronunciación en el idioma francés de los textos mostrados; y, finalmente, se usa un sistema kinestésico al generar sensaciones al usuario a través de los juegos empleados para reforzar el aprendizaje de cada lección.

Gamificación

Según Stieglitz et al. (2017), la gamificación se puede definir como el uso de elementos de diseño de juegos en un contexto ajeno al juego. La gamificación puede ser usada en diferentes productos o servicios, el propósito de su implementación es incrementar el disfrute de dicho producto por parte del usuario y así generar una experiencia agradable, es por esto que la gamificación se ha venido utilizando desde entornos educativos hasta en campañas de marketing.

Elementos del juego. Como se menciona en el informe realizado por el Observatorio de Innovación Educativa del Tecnológico de Monterrey, no hay un acuerdo en la clasificación o descripción de los elementos del juego, sin embargo, se pueden mencionar algunas categorías que se pueden tener en cuenta al momento de llevar a cabo la gamificación.

- Metas y objetivos: generan motivación al presentar un reto o una situación a resolver.
- Reglas: usadas para limitar las acciones de los jugadores y mantener el juego manejable.
- Narrativa: Proporciona un contexto realista.
- Libertad de elegir: Brinda diferentes maneras de alcanzar los objetivos.
- Libertad para equivocarse: Propicia la confianza y participación del estudiante.

- Recompensas: Motivan la competencia y el sentimiento de logro.
- Retroalimentación: Dirige el avance del usuario a partir de su comportamiento.
- Estatus visible: Permite a los participantes tener presente su avance.
- Restricción de tiempo: Puede ayudar a concretar esfuerzos para resolver una tarea.
- Progreso: Permite que el usuario desarrolle habilidades cada vez más complejas.
- Sorpresa: Ayuda a motivar y mantener a los usuarios involucrados.

Metodologías de desarrollo ágil para aplicaciones móviles

El principio de las metodologías ágiles, como señala Amaya (2013), consiste en una mejor captura de los requerimientos cambiantes y en la gestión del riesgo; múltiples metodologías de este tipo se han propuesto con el fin de agilizar los procesos de desarrollo de software tradicionales, y debido a la importancia que ha adquirido el uso de teléfonos inteligentes en la vida diaria, más recientemente estas metodologías ágiles se han enfocado también al desarrollo de aplicaciones móviles.

Metodología mobile-D. Amaya (2013) explica también que una de las principales metodologías ágiles para el desarrollo móvil es Mobile-D, que es usado en equipos de hasta 10 desarrolladores y plantea completar un producto en máximo de 10 semanas, por lo que una de sus principales ventajas es que como afirma Arroyo (2011), se consiguen ciclos de desarrollo muy rápidos en equipos muy pequeños. Sus prácticas incluyen desarrollo basado en pruebas, programación en parejas, integración continua y refactorización; y consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema.

Hybrid methodology design. Según Amaya (2013), esta metodología se basa en un modelo iterativo incremental que permite una entrega rápida de software y una mejor gestión de riesgos; consta de cuatro iteraciones, en la primera, se dividen la fase de análisis y diseño, y se fusionan la fase de implementación y las pruebas; en la segunda iteración, se realiza una integración de partes del modelo New Product Development, se añade generación de ideas al inicio del ciclo y prueba de mercado antes de la fase de comercialización; la tercera iteración integra el motor de desarrollo de los métodos de desarrollo adaptativo, y finalmente, en la cuarta iteración se añaden elementos de prototipado, y se refina la fase de iniciación del proyecto.

Mobile development process Spiral. Se basa en el modelo de desarrollo en espiral, incorporando procesos de evaluación de usabilidad y priorizando la participación del usuario durante todo el proceso; de acuerdo con Nosseir et al. (2012), esta es una metodología apropiada para proyectos grandes, costosos y complicados, pues es modelo destinado a reducción de riesgo. Esta metodología permite a los desarrolladores aplicaciones móviles detallar los criterios de usabilidad de la aplicación, para esto se debe identificar a los usuarios, las tareas y los contextos en los que se utilizará la aplicación, luego se debe dar prioridad a los atributos de usabilidad, identificando cuáles son más importantes para la aplicación y definir métricas que permitan verificar el cumplimiento de cada uno en la aplicación final.

Herramientas Tecnológicas

Android Studio. Es el entorno de desarrollo integrado (IDE) oficial de Android, en el que se puede desarrollar aplicaciones en lenguaje Java o Kotlin; este IDE incluye el editor de código y herramientas para desarrolladores de IntelliJ; en la **Figura 1** se puede observar su logo.

Figura 1

Logo del IDE Android Studio



Nota: Tomado de <https://developer.android.com/studio?hl=es-419>

Spring Boot. Es una herramienta que permite la creación de aplicaciones basadas en Spring, de forma sencilla, proporcionando una estructura básica para el proyecto, ya que realiza una configuración automática para el funcionamiento de Spring Framework. En base al análisis realizado por Haro et al. (2019), esta herramienta presenta importantes características como la posibilidad de desarrollar y arrancar de forma rápida y sencilla aplicaciones basadas en Spring, la realización de configuraciones a través de anotaciones, facilidad para realizar debug en IDE's como Eclipse o IntelliJ, y la disponibilidad de gran cantidad de librerías, lo que representa una ventaja frente a otras herramientas; en la **Figura 2** se puede ver el logo de Spring.

Figura 2

Logo de Spring



Nota: Tomado de <https://spring.io/>

Angular. Angular es un Framework de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una página. Entre sus características se destacan ser multiplataforma, la disponibilidad de herramientas como plantillas para crear rápidamente vistas de interfaz de usuario, la herramienta de línea de comando Angular CLI, y la posibilidad de usar editores e IDEs populares. La plataforma Angular está construida en TypeScript, e incluye un Framework basado en componentes para crear aplicaciones escalables, una colección de librerías bien integradas que incluyen características como enrutamiento, administración de formularios, comunicación cliente-servidor, entre otros; en la **Figura 3** se puede observar su logo.

Figura 3

Logo de Angular

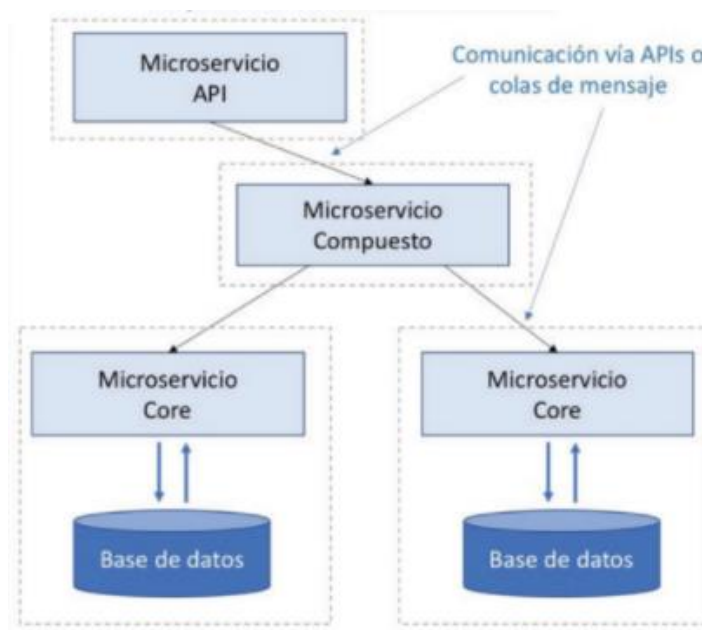


Nota: Tomado de <https://angular.io/>

Microservicios. Como sostienen Roldan et al. (2018), los microservicios se basan en la descomposición de una aplicación en unidades funcionales independientes, para que cada unidad pueda evolucionar sin necesidad de afectar al resto, de igual forma, el testeo, despliegue y otras fases del ciclo de la aplicación, se realizan de forma separada. Entre las principales características de la arquitectura basada en microservicios se encuentran la orientación hacia las capacidades de negocio, la independencia de los microservicios entre sí, la gestión descentralizada de datos y la tolerancia a fallos; la **Figura 4** muestra un esquema de arquitectura basado en microservicios.

Figura 4

Arquitectura basada en microservicios



Nota: Tomado de Roldan et al. (2018)

Pruebas de Software. Las pruebas de software son procesos que garantizan la calidad del software, como lo expone Sommerville (2011), estas pruebas intentan demostrar que un programa hace lo que se intenta que haga y descubrir defectos en el programa antes de usarlo. Para probar un software, este se ejecuta con datos artificiales y se debe verificar los resultados para buscar errores, anomalías o información de atributos no funcionales.

Marco Legal

ISO 21001

Es una norma internacional que establece los requisitos para el uso de un sistema de gestión de organizaciones educativas con el objetivo de mejorar sus procesos y atender todas las necesidades y expectativas de las personas que utilizan sus servicios.

Ley 1341 del 30 de julio de 2009. Artículo 39

Este artículo proporciona información relevante para el uso de las TIC en la educación ya que establece que el Ministerio de Tecnologías de la Información y las Comunicaciones coordinará la articulación del Plan de TIC, con el Plan de Educación y que se brindará apoyo al MEN nacional con estrategias como Fomentar el emprendimiento en TIC, desde los establecimientos educativos, con alto contenido en innovación, capacitar en TIC a docentes en todos los niveles, e incluir la cátedra TIC en todo el sistema educativo, desde la infancia.

Ley 1955 de 2019

Corresponde al Plan Nacional de Desarrollo 2018-2022 y dentro de este existe un pacto por la transformación digital de Colombia. Por lo tanto, uno de los propósitos del Gobierno Nacional en materia TIC es que el país esté conectado al 100%, cerrar la brecha digital, mejorar la participación

en los estratos de menor ingreso y llevar conectividad a estas comunidades a la mayor velocidad posible.

Decreto 393 de 1991

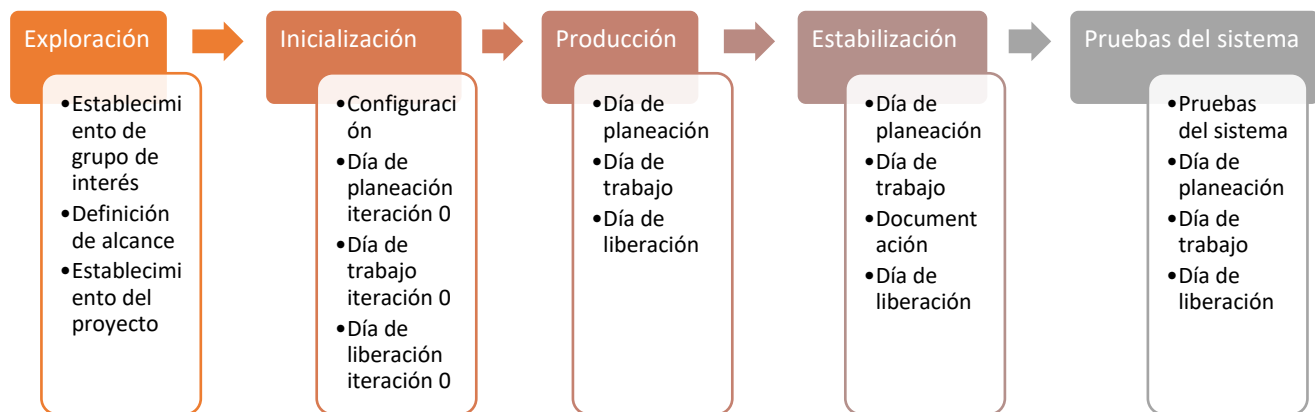
Este decreto ampara las actividades científicas y tecnológicas, proyectos de investigación y creación de tecnologías, de acuerdo a lo dicho en el artículo 1. Así mismo, el artículo 2 del mismo decreto, expone que dentro de los motivos de la asociación se encuentran: Adelantar proyectos de investigación científica, así como también negociar, aplicar y adaptar tecnologías nacionales o extranjeras.

Capítulo 3. Metodología de Desarrollo Ágil

Al elegir una metodología de desarrollo, se deben tener en cuenta dos aspectos claves, que son el tiempo disponible para el desarrollo del proyecto y el tamaño del equipo de trabajo; en base a esto, se ha optado por la metodología Mobile-D, que está diseñada para equipos de trabajo pequeños y su intención es la obtención de un producto final en corto tiempo, ya que tiene ciclos de desarrollo muy rápidos. Esta metodología consta de cinco fases, cada una con sus correspondientes etapas y actividades, como se representa en la **Figura 5**.

Figura 5

Ciclo de desarrollo de la metodología Mobile-D.



Nota: Adaptado de Ciclo de desarrollo Mobile-D, de Amaya, 2013.

Fase de Exploración

Es la primera fase de la metodología, en esta se deben establecer las características del proyecto. En la fase de exploración se fijó el alcance del proyecto; para esto, se definieron los requisitos y acotaciones del sistema, así como los casos de uso.

Inicialización

En esta fase se realizaron actividades de diseño y planificación del proyecto; se realizó un esquema de la arquitectura del proyecto y se definieron las tecnologías que usarían en cada componente, posteriormente se definió el contenido programático de las lecciones que se incluirían en la aplicación, se diseñaron las interfaces de la aplicación móvil y se realizó el modelado de las bases de datos.

Producción

Esta fase se lleva a cabo de forma iterativa, siguiendo tres etapas, planeación, ejecución y liberación, para cada componente del sistema, hasta que se haya completado el mismo. En esta fase se realizó el desarrollo tanto de backend como frontend del proyecto, se desarrollan además las pruebas unitarias de cada funcionalidad de la aplicación.

Estabilización

Esta fase corresponde a actividades de integración del proyecto, así como también de documentación; en este caso, se realizó la integración del backend con el frontend, mediante el consumo de servicios, y se desarrolló la documentación correspondiente, incluyendo manuales de usuario y administrador.

Pruebas del Sistema

La fase de pruebas del sistema es la última en la metodología y el objetivo de esta es entregar una versión estable y completamente funcional del sistema; para esto se realizaron pruebas de la aplicación terminada, en base a los requerimientos establecidos, y se realizaron los ajustes necesarios para solventar cualquier error presentado.

Capítulo 4. Desarrollo del Sistema

Fase de Exploración

Con el fin de establecer el alcance del proyecto, se definieron los requerimientos funcionales y no funcionales, tomando como referencia el estándar IEEE830 para especificación de requisitos de software, y se establecieron, además, las acotaciones del proyecto referentes tanto a desarrollo, y requisitos del dispositivo móvil, como contenido temático de francés; se realizó también una comparación de herramientas de tecnologías de la información y las comunicaciones usadas para el aprendizaje.

Requisitos Comunes de los Interfaces

Interfaces de usuario. La paleta de colores empleada en la app debe basarse en los colores de la bandera de Francia, ajustando los tonos para que permitan una correcta visualización del contenido y sea agradable a la vista.

No se empleará color negro para el texto, sino diferentes tonos de gris, para evitar contrastes fuertes que pueden ser molestos a la vista del usuario.

Requisitos de Hardware. Al ser una aplicación móvil, se requiere de un smartphone para su ejecución.

La app usará el reproductor de audio del dispositivo.

Requisitos de software. La aplicación se desarrollará para el sistema operativo Android.

Requisitos Funcionales

Requisito funcional 1

Tabla 1

Requisito funcional 1

Número de requisito	RF01
Nombre de requisito	Registro de usuarios
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	Se debe tener un registro de los usuarios para llevar un control adecuado de su avance en cada lección.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

El sistema debe permitir el registro de usuarios mediante los datos nombre, correo electrónico y contraseña, que serán ingresados por el mismo, y un id que se asignará automáticamente.

Requisito funcional 2

Tabla 2

Requisito funcional 2

Número de requisito	RF02
Nombre de requisito	Inicio de sesión
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	El inicio de sesión permite que el usuario acceda a las lecciones.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

El sistema debe permitir el inicio de sesión a los usuarios que se hayan registrado previamente, mediante la validación de su correo electrónico y contraseña.

Requisito funcional 3

Tabla 3

Requisito funcional 3

Número de requisito	RF03
Nombre de requisito	Pantalla de contenido
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	Es necesario tener una forma de navegar rápidamente entre las diferentes lecciones de la app.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

Se debe presentar una pantalla de contenido accesible desde cualquier pantalla de la app, que muestre los nombres de cada lección y permita acceder a cada una de ellas.

Requisito funcional 4

Tabla 4

Requisito funcional 4

Número de requisito	RF04
Nombre de requisito	Pantallas de lecciones
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	La pantalla de lección es necesaria para que el usuario reciba la información de cada tema.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

Para cada tema, se debe presentar una pantalla de lección, que consiste en la teoría o explicación del mismo.

Requisito funcional 5

Tabla 5

Requisito funcional 5

Número de requisito	RF05
Nombre de requisito	Reproducción de audio
Tipo	☒ Requisito ☐ Restricción

Fuente del requisito	La reproducción de audio permite al usuario escuchar la pronunciación correcta del vocabulario en francés.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

En las lecciones se incluirán audios con la pronunciación de palabras o frases, para que el usuario reproduzca.

Requisito funcional 6

Tabla 6

Requisito funcional 6

Número de requisito	RF06
Nombre de requisito	Actividades interactivas
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	Las actividades interactivas permiten afianzar lo aprendido en las lecciones.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

Para cada tema, se debe incluir una o más actividades de práctica en la que el usuario interactúe con la app.

Requisito funcional 7

Tabla 7

Requisito funcional 7

Número de requisito	RF07
Nombre de requisito	Ingreso de texto
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	El ingreso de texto permite que el usuario practique la escritura del idioma.
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

La app debe permitir al usuario ingresar texto para resolver las actividades de práctica que se presenten en las lecciones.

Requisito funcional 8

Tabla 8

Requisito funcional 8

Número de requisito	RF08
Nombre de requisito	Botones de selección
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	Seleccionar la opción correcta permite al usuario practicar el idioma mediante el reconocimiento de vocabulario.
Prioridad del requisito	☒ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional

La app debe proporcionar al usuario botones para seleccionar la respuesta correcta en las actividades de práctica que lo requieran.

Acotaciones

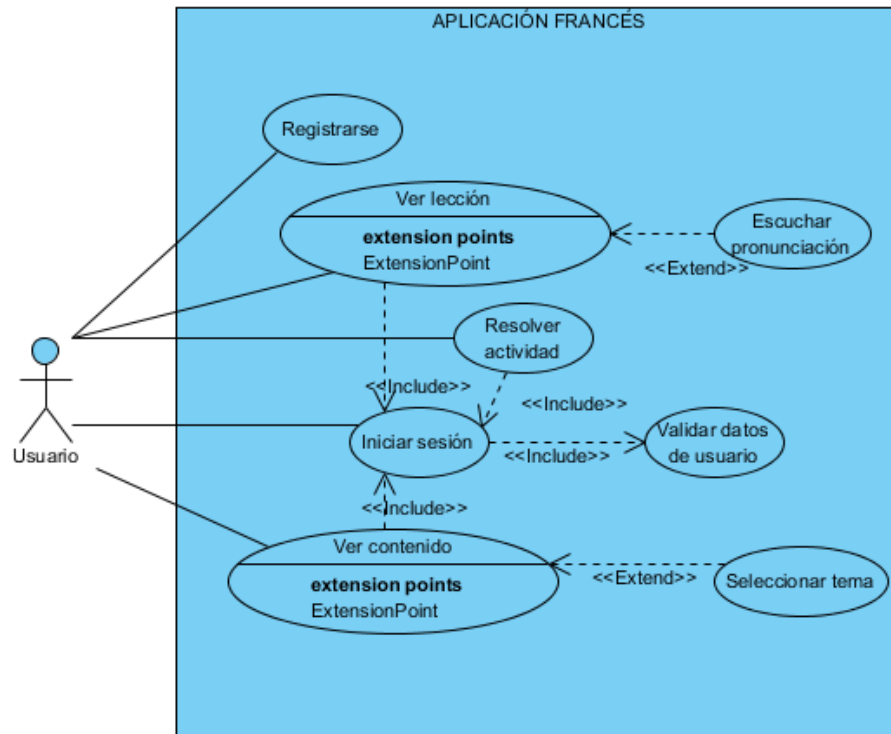
- La aplicación se desarrollará únicamente para dispositivos con sistema Android, versión mínima 4.1 (Jelly Bean).
- Se abarcarán temáticas de francés correspondiente exclusivamente al nivel A1 según el Marco Común Europeo de Referencia para las Lenguas (MCERL).
- La metodología de desarrollo a utilizar para la aplicación móvil es Mobile-D.
- Para el desarrollo del Backend se usará la herramienta Spring Boot.
- El Frontend se compone de dos partes correspondientes a la aplicación móvil, para la cual no se empleará un Framework y una plataforma administrativa en la cual se usará Angular.

Diagrama de Casos de Uso

En la figura se muestra el diagrama de casos de uso de la aplicación.

Figura 6

Diagrama de casos de uso



Comparación de herramientas TIC

En esta sección se comparan cuatro herramientas de tecnologías de la información y las comunicaciones usadas tanto en el aprendizaje de idiomas como en el aprendizaje en general, para esto se tomaron en cuenta aspectos como la posibilidad de presentar información detallada usando ese tipo de herramienta, la posibilidad de proporcionar contenido personalizado, la posibilidad de realizar participación colaborativa durante el aprendizaje, el tiempo que se suele requerir para estudiar un tema o una clase mediante cada herramienta, el nivel de portabilidad que proporciona la herramienta, la posibilidad de inmediatez y el entretenimiento que brinda el uso de la herramienta, en la **Tabla 9**, se muestra esta comparación.

Tabla 9

Comparación de herramientas de Tecnologías de la Información y las Comunicaciones para el aprendizaje

	M-Learning	E-Learning	Social Software	Videojuegos
Información detallada	Se sintetiza la información para adaptarse a la pantalla de los dispositivos	Recursos educativos amplios con información muy detallada	Puede presentar gran cantidad de información	La información se presenta de forma didáctica
Contenido personalizado	El contenido se encuentra establecido en la aplicación	El contenido se encuentra establecido en el curso	Se puede crear contenido solicitado por los miembros	El contenido se encuentra establecido en el juego
Participación colaborativa	Puede emplear participación colaborativa, no es el enfoque principal	Puede emplear participación colaborativa, no es el enfoque principal	Es el enfoque principal	Puede emplear participación colaborativa, no es el enfoque principal
Tiempo requerido	El contenido se diseña para presentar cada tema en corto tiempo	Requiere gran cantidad de tiempo al presentar contenidos amplios	Se puede diseñar contenido para poco o bastante tiempo	El tiempo requerido puede variar
Portabilidad	Se puede transportar a cualquier lugar con facilidad	Puede ser portable si se usa computador portátil	Depende del tipo de dispositivo desde el cual se acceda	Depende del tipo de dispositivo para el cual se diseñe el juego
Inmediatez	Se puede acceder con inmediatez a las aplicaciones	Depende del dispositivo desde el cual se acceda	Depende del dispositivo desde el cual se acceda	Depende del dispositivo desde el cual se acceda
Entretenimiento	Puede emplear elementos de entretenimiento	Puede emplear elementos de entretenimiento	Se puede diseñar para proporcionar entretenimiento	Su enfoque principal es el entretenimiento

Fase de Inicialización

Contenido temático

El contenido temático de la aplicación consiste en diez lecciones correspondientes a temas de francés nivel A1 según el Marco Común Europeo de Referencia para las Lenguas (MCERL), como se ven a continuación:

Salutations: Saludos

Bonjour: Buenos Días

Bonsoir: Buenas Tardes

Bonne Nuit: Buenas Noches (Despedida)

Salut: Hola

Au Revoir: Adiós, Hasta Luego

Ça Va?: ¿Cómo Estás? (Informal)

Comment Allez-Vous?: ¿Cómo Está Usted?

Comment Vas-Tu: ¿Cómo Estás Tú?

Bien: Bien

Ça Va: Bien

Très Bien: Muy Bien

Mal: Mal

Très Mal: Muy Mal

Pas Trop Mal: No Muy Mal

Comme Si Comme Ça: Más O Menos (Informal)

Pus Ou Moins: Más O Menos

Et Vous?: ¿Y Usted ?

Et Toi: ¿Y Tú ?

Comment Vous Appelez-Vous?: ¿Cómo Se Llama Usted ?

Comment Vous Vous Appelez?: ¿Cómo Se Llama Usted ?

Comment T'appelles-Tu ?: ¿Cómo Te Llamas Tú ?

Comment Tu T'appelles ?: ¿Cómo Te Llamas Tú ?

Je M'appelle ... : Yo Me Llamo

Je Suis ...: Yo Soy

Monsieur: Señor

Madame: Señora

Mademoiselle: Señorita

Enchanté: Encantado

À Bientôt: Hasta Pronto

Bienvenue: Bienvenido

Alphabet: Alfabeto

A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

Nombres: Números

Un: Uno

Deux: Dos

Trois: Tres

Quatre: Cuatro

Cinq: Cinco

Six: Seis

Sept: Siete

Huit: Ocho

Neuf: Nueve

Dix: Diez

Onze: Once

Douze: Doce

Treize: Trece

Quatorce: Catorce

Quinze: Quince

Seize: Dieciseis

Dix-Sept: Diecisiete

Dix-Huit: Dieciocho

Dix-Neuf: Diecinueve

Vingt: Veinte

Couleurs: Colores

Noir/Noire: Negro

Gris/Grise: Gris

Blanc/Blanche: Blanco

Bleu/Bleue: Azul

Violet/Violette: Violeta

Vert/Verte: Verde

Jaune: Amarillo

Orangé: Naranja

Marron: Marrón

Rouge: Rojo

Rose: Rosa

Beige: Beige

Jours de la semaine: Días de la semana

Lundi: Lunes

Mardi: Martes

Mercredi: Miercoles.

Jeudi: Jueves

Vendredi: Viernes

Samedi: Sabado

Dimanche: Domingo

Mois de l'année: Meses del año

Janvier: Enero

Février: Febrero

Mars: Marzo

Avril: Abril

Mai: Mayo

Juin: Junio

Juillet: Julio

Août: Agosto

Septembre: Septiembre

Octobre: Octubre

Novembre: Noviembre

Decembre: Diciembre

Les Saison: Las estaciones

Le Printemps: Primavera

L'été: Verano

L'automne: Otoño

L'hiver: Invierno

8. Quelques Expressions En Français

Il Fait Chaud: Hace Calor

Il Y A Du Soleil: Hay Sol

Il Fait Frais: Hace Freco

Il Y A Des Nuages: Hay Nubes

Il Fait Froid: Hace Frio

Il Y A Du Vent: Hay Viento

Il Pleut: Llueve

Pronoms personnels: Pronombres personales

Je: Yo

Tu: Tú

Il : Él

Elle: Ella

On: Nosotros

Nous: Nosotros

Vous: Ustedes

Ils: Ellos

Elles: Ellas

Verbe Être: Ser/Estar

Je Suis: Yo Soy

Tu Es: Tú Eres

Il, Elle, On Est: Él/Ella Es

Nous Sommes: Nosotros Somos

Vous Êtes: Ustedes Son

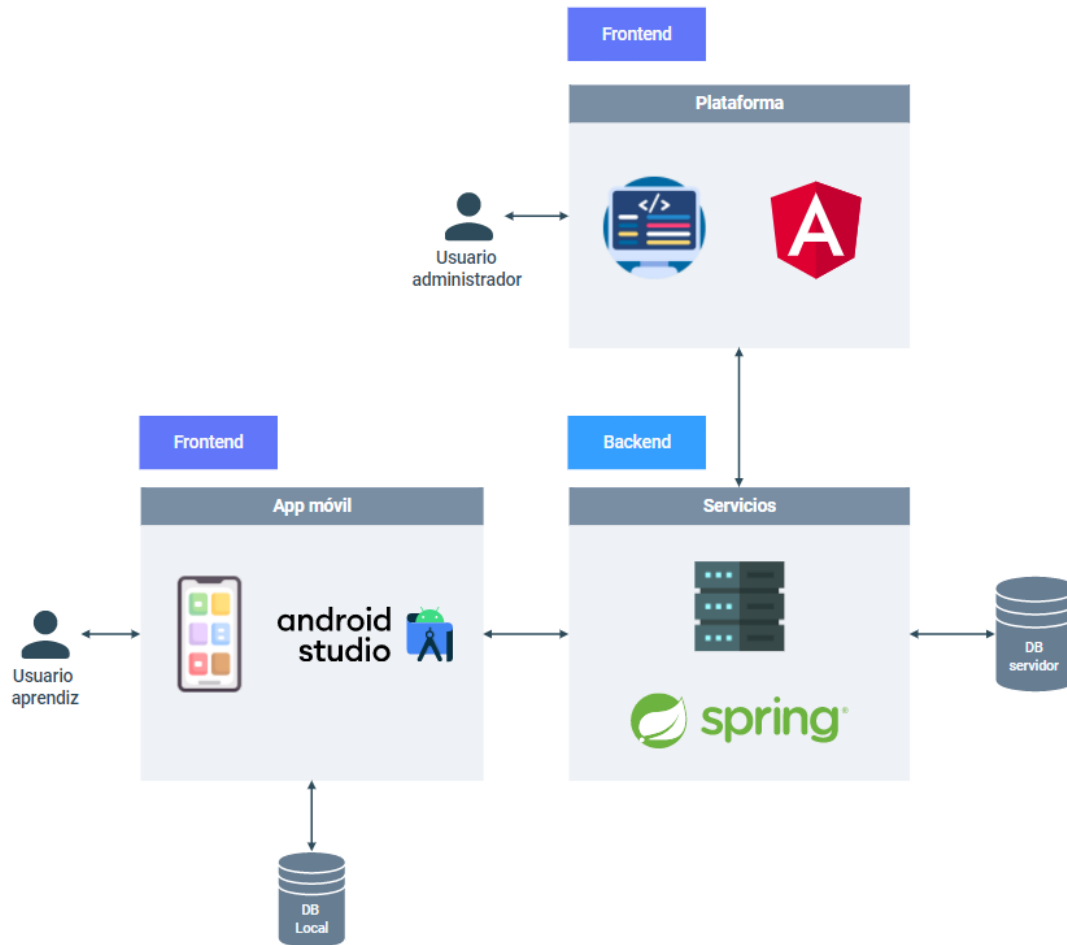
Ils, Elles Sont: Ellos/Ellas Son

Esquema de arquitectura

El sistema se compone esencialmente de tres partes, el backend que corresponde a los servicios, el cual se comunica directamente con la base de datos del servidor y dos partes de frontend, correspondientes a la plataforma administrativa y la aplicación móvil, la cual cuenta con una base de datos local usada para almacenar la información de las lecciones, para que, de esta forma, la aplicación pueda funcionar cuando no se cuenta con conexión a internet. Para visualizar y comprender el funcionamiento del proyecto completo, se realizó un esquema de arquitectura que se muestra en la **Figura 7**.

Figura 7

Esquema de arquitectura volilá



Diseño de interfaces

En esta fase se realizó el diseño de las interfaces básicas que tendría la aplicación, incluyendo la pantalla de inicio de sesión, registro de usuario, lección y menú de contenido. Ver figuras 9, 10, 11 y 12.

Figura 8

Bosquejo de pantalla de inicio de sesión

A sketch of a mobile application login screen. The screen is enclosed in a rounded rectangle representing a smartphone. At the top, there is a status bar with a signal strength indicator, a Wi-Fi icon, and the time 4:54. Below the status bar, there are two input fields: the first is labeled "Correo electrónico" and the second is labeled "Contraseña" with a small eye icon to its right. Below these fields is a button labeled "Ingresar". At the bottom of the screen, there is another button labeled "Registrarse".

Figura 9

Bosquejo de pantalla de registro de usuario

A sketch of a mobile application user registration screen. The screen is enclosed in a rounded rectangle representing a smartphone. At the top, there is a status bar with a signal strength indicator, a Wi-Fi icon, and the time 10:57. Below the status bar, there are four input fields: the first is labeled "Nombres", the second is labeled "Correo electrónico", the third is labeled "Contraseña" with a small eye icon to its right, and the fourth is labeled "Confirmar contraseña" with a small eye icon to its right. Below these fields is a button labeled "Registrarse".

Figura 10

Bosquejo de pantalla de lección

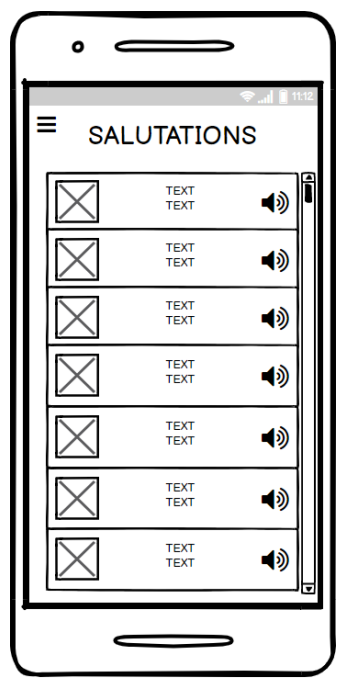
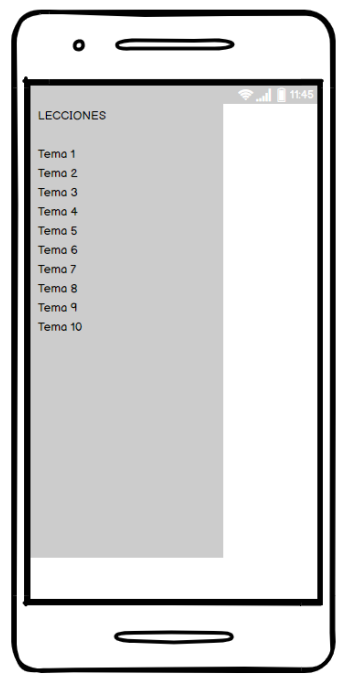


Figura 11

Bosquejo de menú de lecciones



Diseño de base de datos

Debido a las necesidades del proyecto, se requiere diseñar dos bases de datos, la primera, cuyo diseño se muestra en la **Figura 12**, corresponde a la base de datos remota que almacena información sobre todos los usuarios y datos sobre el uso que cada uno de ellos le da a la aplicación móvil; esta es accedida mediante los servicios creados en el Backend del sistema, que son consumidos desde la plataforma administrativa para consultar la información que se requiera, y desde la aplicación móvil para hacer registro de usuarios y actualizar información sobre el avance en las lecciones, cuando se cuente con conexión a internet; la otra base de datos, mostrada en la **Figura 13**, corresponde a una base de datos local, alojada en el dispositivo móvil, que permite que la aplicación funcione sin internet, manteniendo la sesión iniciada y guardando avances de lecciones a medida que se use la aplicación.

Figura 12

Modelo relacional base de datos remota

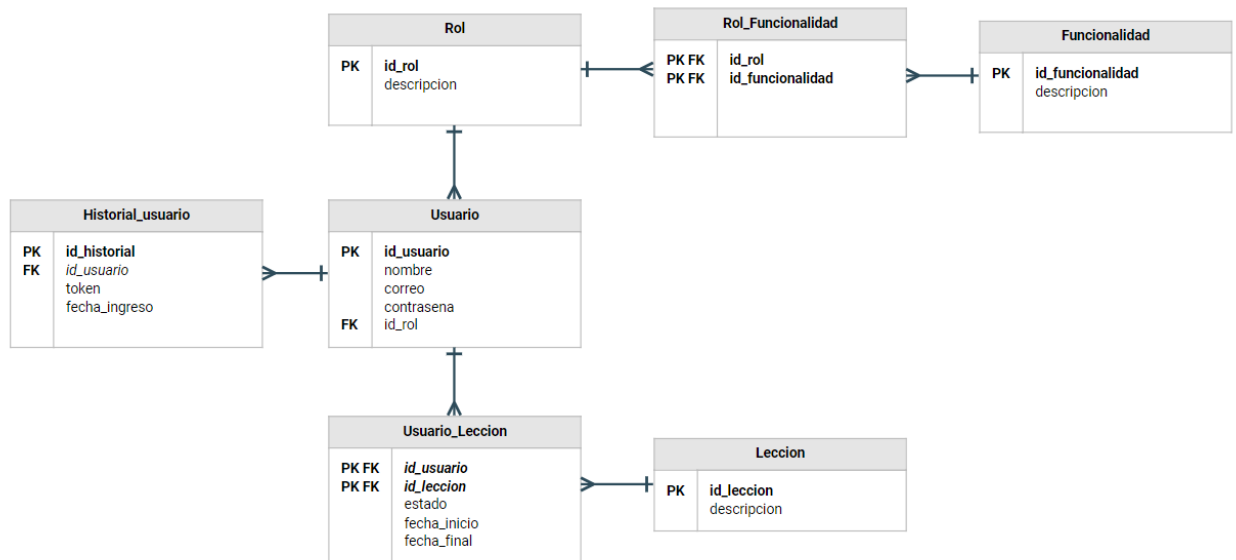
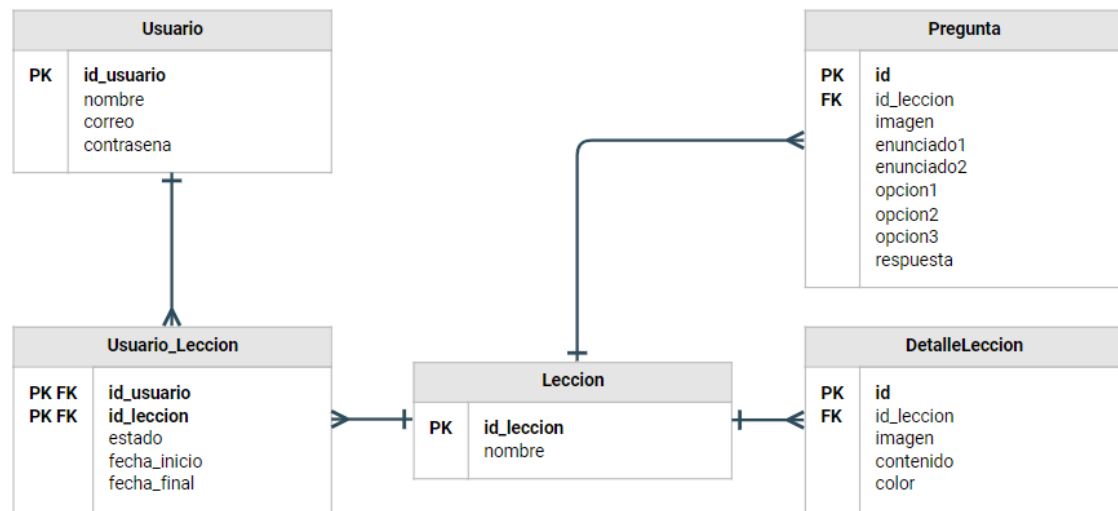


Figura 13

Modelo relacional base de datos local



Fase de Producción

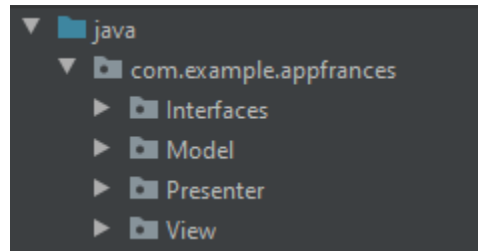
Desarrollo de aplicación móvil

La aplicación móvil fue desarrollada para el sistema operativo Android, por lo que se empleó el IDE Android Studio, y se programó usando lenguaje Java; se usó, además, como se explicará más adelante, el lenguaje de marcado XML.

Arquitectura MVP. El desarrollo de la aplicación móvil se realizó empleando el patrón de arquitectura MVP (Modelo – Vista - Presentador), para lo cual se crearon los respectivos paquetes en Android Studio, y en cada uno de ellos se crearon las clases java que pertenecen a cada capa, además de un paquete “interfaces” en el que se crearon las interfaces Java necesarias, como se muestra en la **Figura 14**.

Figura 14

Paquetes MVP en Android Studio



Interfaces gráficas. Para el diseño de cada una de las pantallas de la aplicación, se crearon layouts en el paquete que Android Studio proporciona para tal fin, como se ve en la **Figura 15**. Para crear cada uno de los componentes de la interfaz gráfica y asignar sus respectivas características, se emplea lenguaje xml, y mediante las herramientas del IDE se puede tener una vista previa de la pantalla, como se vería en un dispositivo real, ver **Figura 16**.

Figura 15

Paquete de layout en Android Studio.

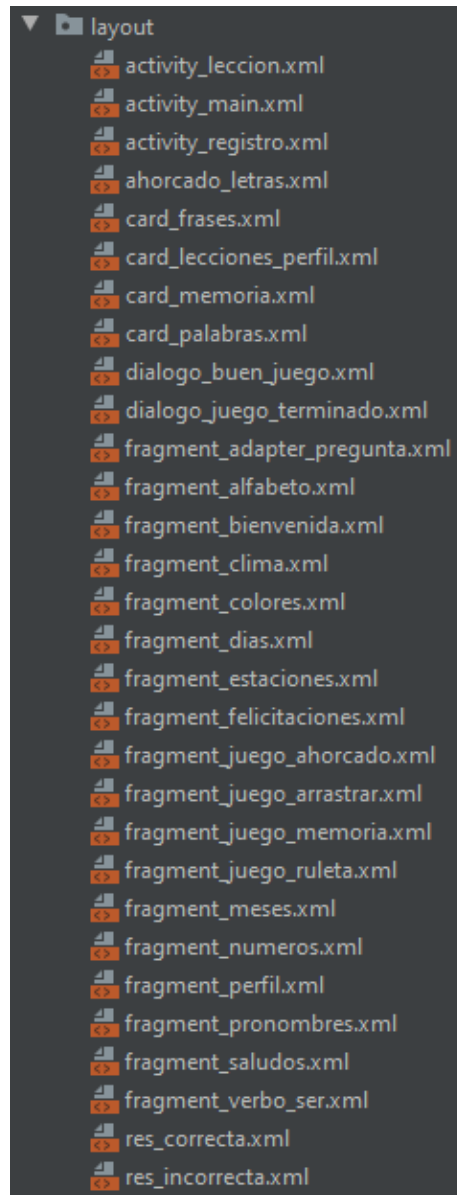
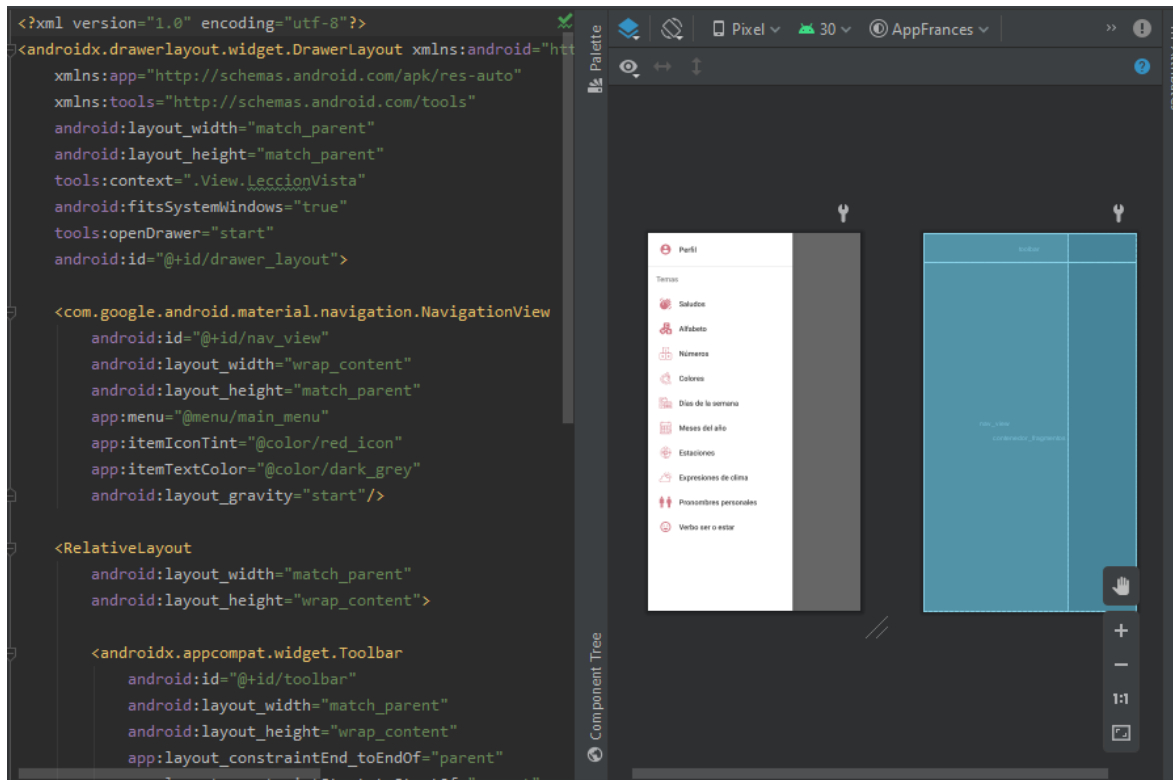


Figura 16

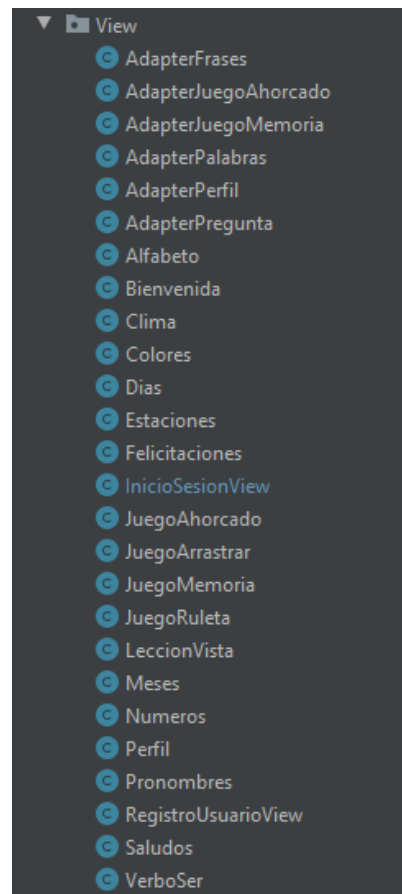
Código XML y vista previa de interfaz gráfica



A cada una de las interfaces creadas con xml, se le dio funcionalidad usando las clases Java ubicadas en el paquete “View” que corresponde a la capa de vista, ver **Figura 17**; dado que esta capa se encarga de mostrar información al usuario en la pantalla del dispositivo, se encuentran aquí los Activities, Fragments y Adapters usados en toda la aplicación.

Figura 17

Clases Java pertenecientes a la capa de vista.



Durante el diseño del componente gráfico de la aplicación, uno de los primeros elementos que se configuró fue el ícono con el cual la aplicación se mostraría en el dispositivo móvil, para lo cual se importó un vector con extensión .svg y se realizaron las configuraciones necesarias en Android Studio para finalmente obtener el ícono que se muestra en la **Figura 18**.

Figura 18

Ícono de app Voilà en el dispositivo móvil.



En la sección de interfaces gráficas se emplearon también animaciones que fueron agregadas en formato json, como se muestra en la **Figura 19**, y posteriormente fueron configuradas para ser mostradas en cuadros de diálogo al finalizar un juego en la sección de práctica, dependiendo si el juego fue ganado, ver **Figura 20**, o perdido, ver **Figura 21**.

Figura 19

Archivos de animaciones en formato json

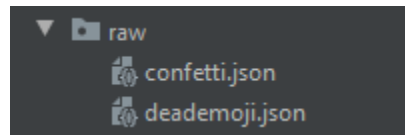


Figura 20

Cuadro de diálogo con animación para juego ganado

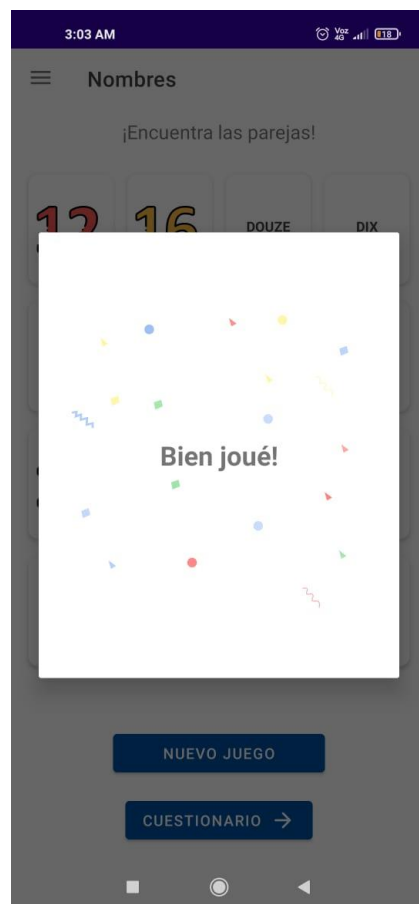
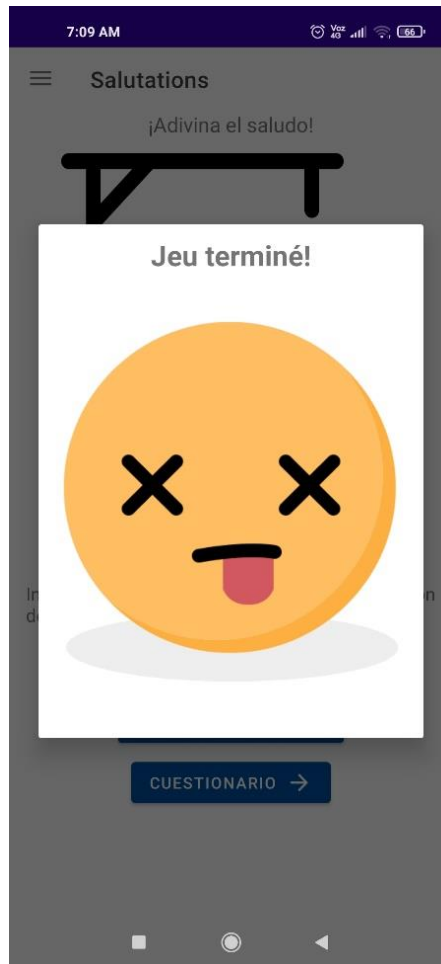


Figura 21

Cuadro de diálogo con animación para juego perdido



Algunas de las imágenes utilizadas en la aplicación fueron obtenidas de forma gratuita en el sitio web de Flaticon, mientras que otras fueron diseñadas empleando el software de diseño gráfico Adobe Illustrator, por otro lado, las animaciones fueron obtenidas del sitio web de LottieFiles.

Desarrollo de las lecciones. Las diez lecciones de la aplicación se presentan usando dos tipos de diseño, dependiendo de cuál se adaptara más al contenido de la lección; el primer diseño, mostrado en **Figura 22**, consiste en tarjetas que ocupan todo el ancho de la pantalla cada una, y se componen de una imagen, un texto en francés, texto en español y un ícono de audio con el cual se puede escuchar la pronunciación del texto en francés, este tipo de diseño se usó en las lecciones donde el texto que debía mostrarse eran frases de varias palabras; el segundo diseño, mostrado en **Figura 23**, consiste en tarjetas las cuales se distribuyen dos en cada fila, y se usó en las lecciones donde el texto a mostrar era corto, la reproducción de audio en este tipo de diseño, se da presionando cada tarjeta.

Figura 22

Lección Saludos

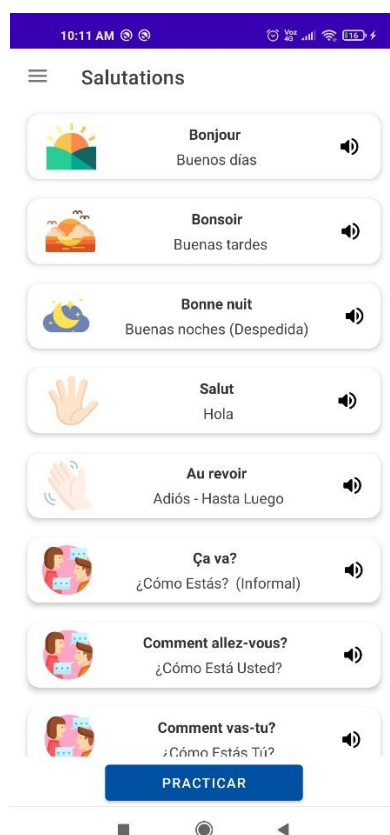


Figura 23

Lección Números



Para los dos tipos de diseño se emplearon elementos dinámicos que permiten crear el diseño de una sola tarjeta como se ve en la **Figura 24** y por medio de una clase adaptador, cargar la información necesaria empleando listas como se puede observar en la **Figura 25**, donde se muestra el constructor del adaptador para el diseño de la **Figura 22**, el cual recibe un parámetro lista que contiene toda la información que se debe mostrar en las tarjetas; lo anterior permite la escalabilidad de la aplicación, ya que podría añadirse más elementos de forma muy rápida en caso de querer ampliar la información de las lecciones en el futuro, además, al emplear la misma clase adaptador y el mismo

layout, para todas las lecciones que usan ese tipo de diseño, se garantiza la reutilización y por lo tanto, se evita la duplicidad de código.

Figura 24

Diseño tarjeta para mostrar contenido de lecciones

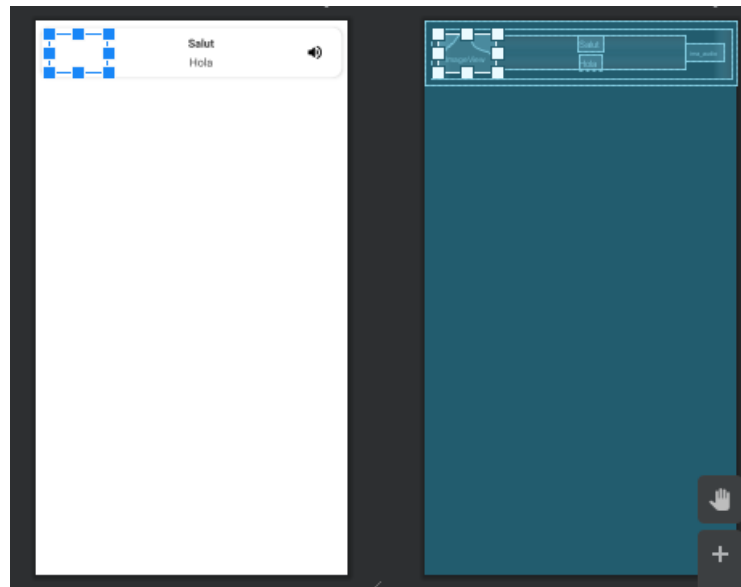


Figura 25

Constructor de clase adaptador

```
public class AdapterFrases extends RecyclerView.Adapter<AdapterFrases.ViewHolder> {  
  
    List<DetalleLeccion> detalleLeccionList;  
    LayoutInflater inf;  
  
    public AdapterFrases(Context context, List<DetalleLeccion> detalleLeccionList) {  
        this.detalleLeccionList = detalleLeccionList;  
        this.inf=LayoutInflater.from(context);  
    }  
}
```

Pronunciación. Todas las lecciones tienen la opción de reproducir el texto que se muestra en francés, para esto, se usó la clase TextToSpeech de Android, el cual permite en su inicialización, seleccionar un lenguaje o una región, para configurar el audio con su acento, como se ve en la **Figura 26**, para la aplicación Voilà se configuró el lenguaje especificando el país Francia.

Figura 26

Inicialización de TextToSpeech

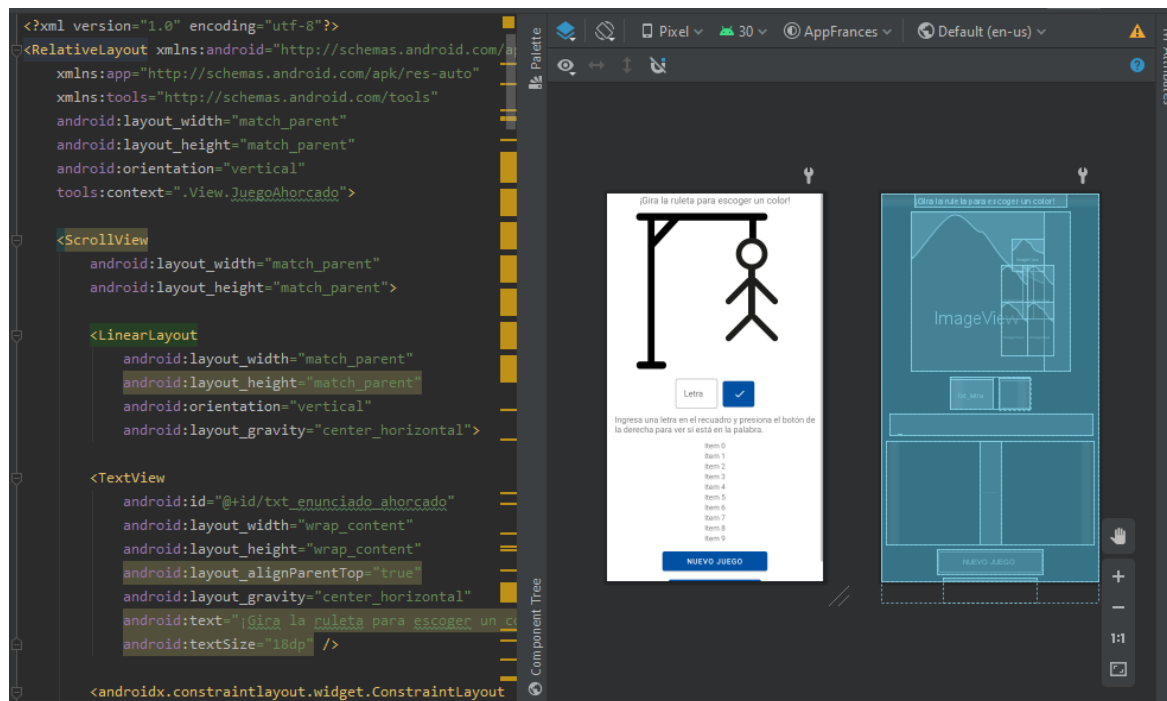
```
tts = new TextToSpeech( context: this, new TextToSpeech.OnInitListener() {  
    @Override  
    public void onInit(int status) {  
        if(status == TextToSpeech.SUCCESS){  
            int result = tts.setLanguage(Locale.FRANCE);  
            if(result == TextToSpeech.LANG_MISSING_DATA || result == TextToSpeech.LANG_NOT_SUPPORTED){  
                Log.e( tag: "tts", msg: " no soportado");  
            }else{  
                Log.e( tag: "tts", msg: " si soportado");  
            }  
        }else{  
            Log.e( tag: "tts", msg: " no soportado");  
        }  
    }  
});
```

Gamificación. Para la sección de práctica de la aplicación móvil, se diseñaron 4 tipos de juegos que fueron empleados en las diferentes lecciones dependiendo de cuál se ajustara mejor a los contenidos de la lección. Cada uno de los juegos se desarrolló en Android Studio diseñando layouts xml y clases java en forma general de manera que el contenido de cada tema pudiera cargarse de forma dinámica en la pantalla.

El primer juego implementado fue el juego de ahorcado, que consiste en descifrar las letras que corresponden a la palabra o frase oculta, en una cantidad definida de intentos; su diseño se muestra en la **Figura 27**.

Figura 27

Diseño de juego ahorcado.



En la **Figura 28** se puede observar una captura de pantalla de la aplicación correspondiente al juego usado en la lección “saludos”; este juego fue empleado también para las lecciones “días de la semana” y “verbo ser o estar”.

Figura 28

Juego ahorcado en lección saludos.



El siguiente juego implementado corresponde a un juego de memoria, el cual fue usado únicamente para la lección “números” ya que se adapta adecuadamente a este tema, pues permite mostrar en una tarjeta el número y en otra la forma de escribir el número en francés, con el fin de encontrar las tarjetas que hagan pareja; sin embargo, al igual que los demás juegos, fue diseñado para adaptar su contenido de forma dinámica. En la **Figura 29** se puede observar el diseño del juego y en la **Figura 30** se muestra una captura de pantalla de la aplicación.

Figura 29

Diseño de juego memoria

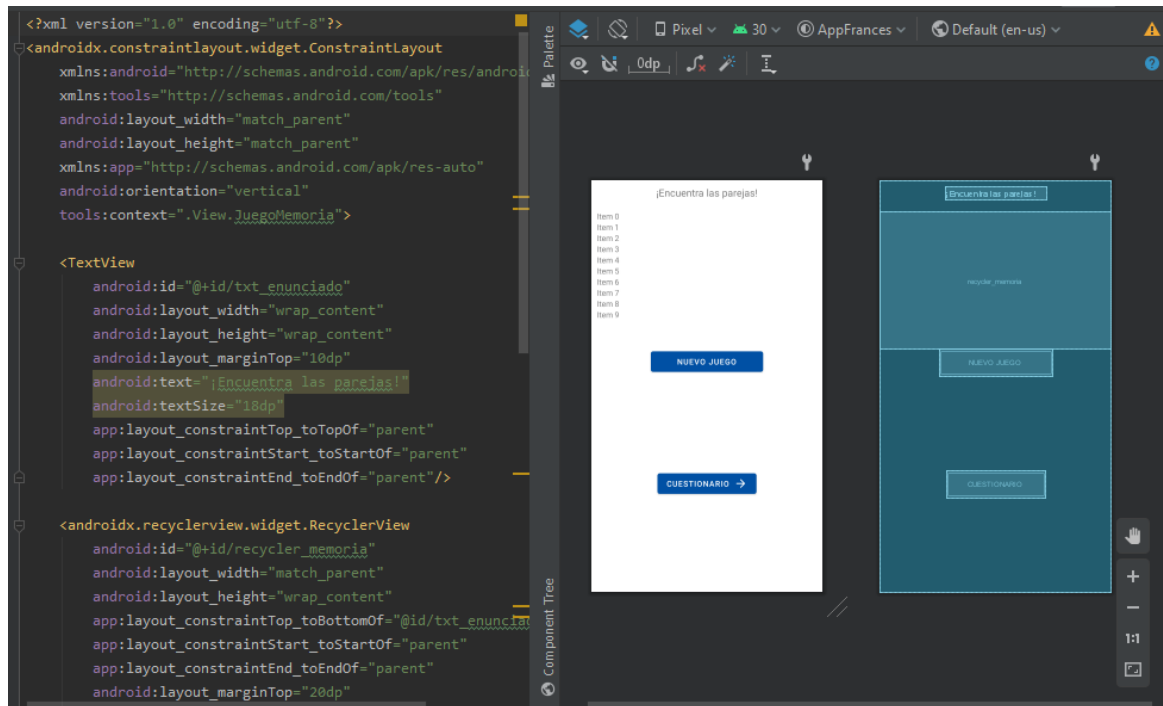
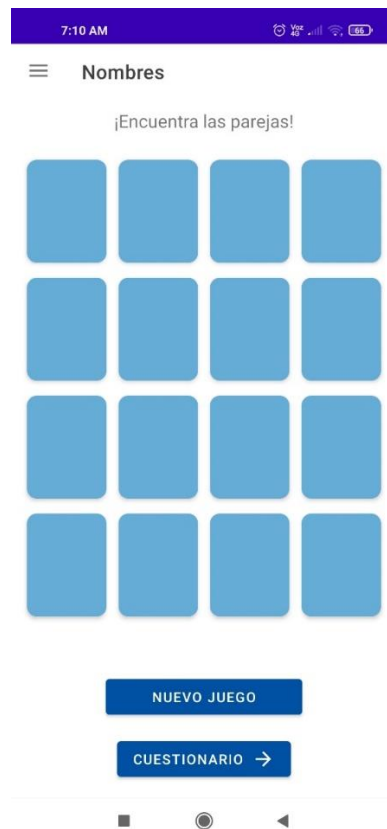


Figura 30

Juego memoria en lección números.



El siguiente juego fue una ruleta, que consiste en hacer girar la ruleta y escribir la palabra en francés correspondiente a la sección que la ruleta indica; este juego fue utilizado para las lecciones “colores” y “meses del año”. La **Figura 31** muestra el diseño del juego de ruleta y la **Figura 32** muestra una captura de pantalla de la aplicación para la lección “colores”.

Figura 31

Diseño juego ruleta.

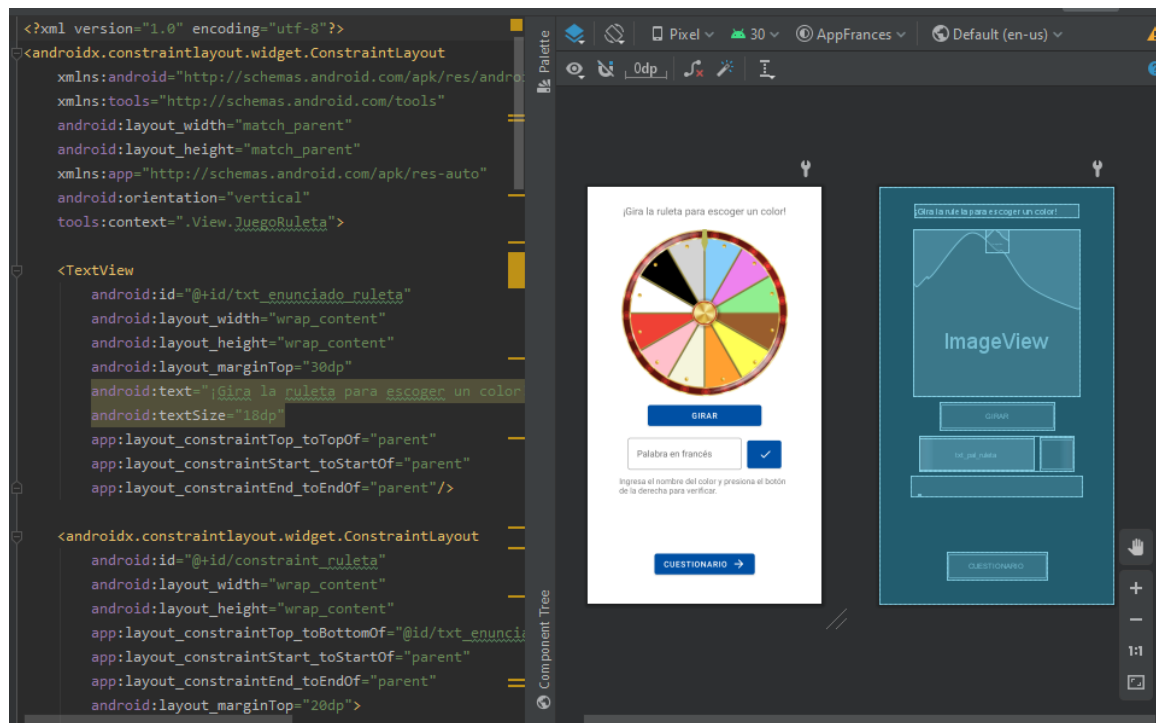


Figura 32

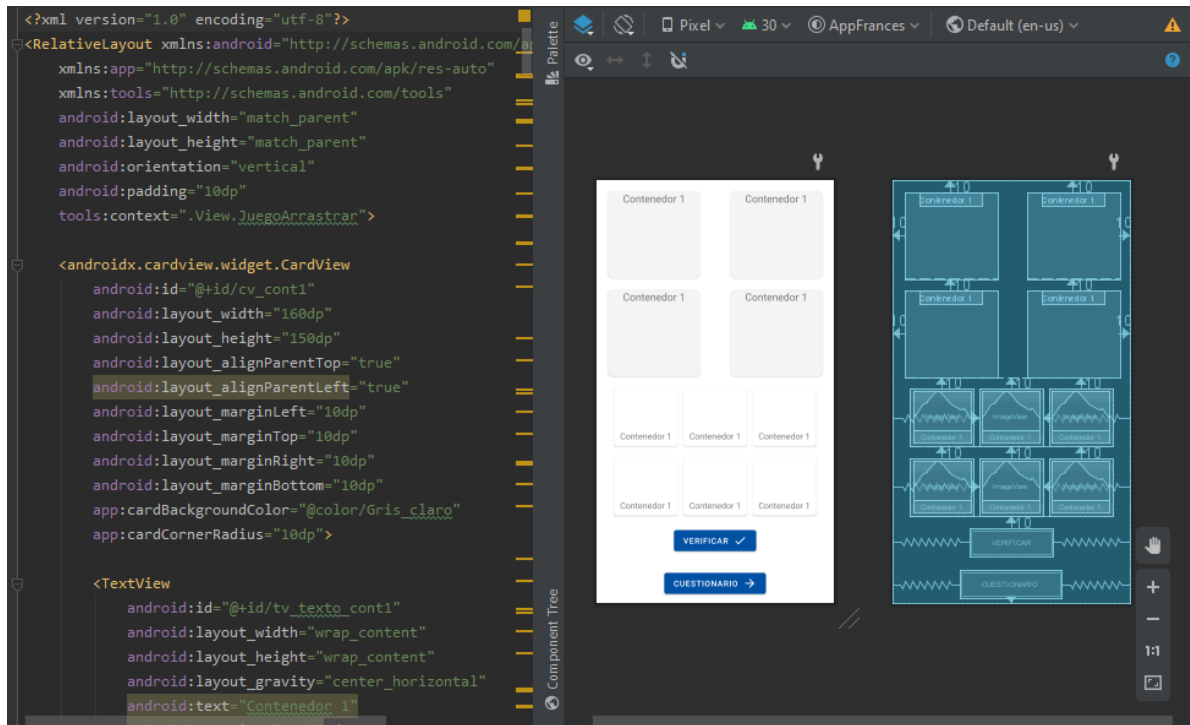
Juego ruleta en lección colores.



El último juego consiste en arrastrar unas tarjetas a los recuadros correspondientes, cada recuadro contiene un texto en español, y cada tarjeta contiene una imagen y un texto en francés que debe coincidir con la traducción ubicada en el recuadro; en la **Figura 33** se puede observar el diseño de este juego.

Figura 33

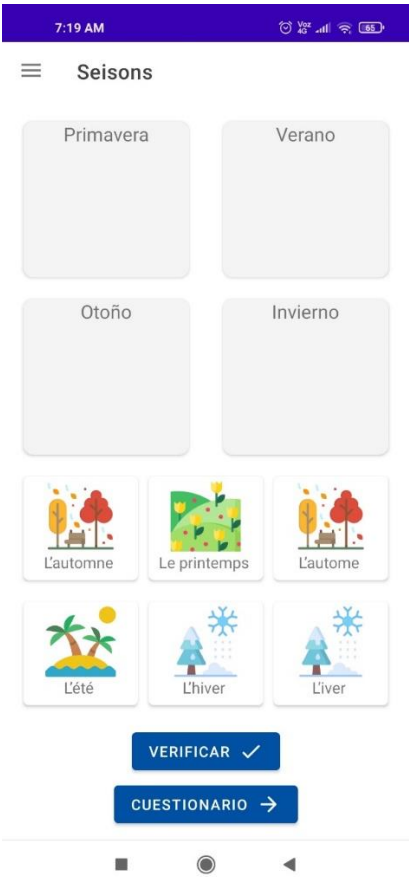
Diseño de juego de arrastrar.



El juego de arrastrar tarjetas fue usado en las lecciones “estaciones”, “expresiones de clima”, y “pronombres personales”. En la **Figura 34** se muestra una captura de pantalla del juego correspondiente a la lección “estaciones”.

Figura 34

Juego arrastrar en lección estaciones.



Pruebas unitarias y de integración. Para verificar el funcionamiento de los métodos de la aplicación, se realizaron pruebas unitarias y pruebas de integración empleando el framework de testing Junit. En la **Figura 35** se muestra una prueba unitaria en la que, por medio de la anotación `@Parameters`, se validan tres posibles escenarios de correo y sus respuestas esperadas, la primera cadena de texto, corresponde a un correo electrónico que cumple con el patrón exigido, por lo que su respuesta debe ser verdadero o "true", en la segunda cadena de texto, el correo no cumple con el patrón ya que no contiene el símbolo `@`, por lo que la respuesta debe ser falso o "false", y en la tercera cadena de texto, el correo no contiene terminación de dominio válida, es decir, que finalice con un punto seguido de caracteres (como `.com`, `.co`, `.edu.co`), por lo que su respuesta también debe ser falso o "false"; y en la **Figura 36** se puede ver que la prueba pasó los test ya que se produjeron los resultados que se esperaban.

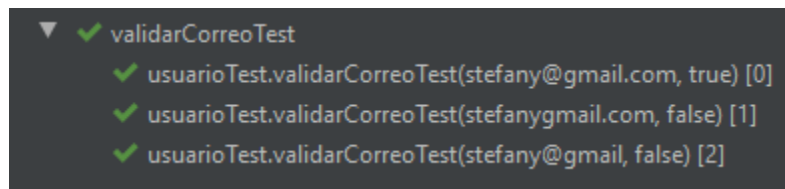
Figura 35

Prueba unitaria para validar correo

```
@Test
@Parameters({"stefany@gmail.com, true",
            "stefanygmail.com, false",
            "stefany@gmail, false" })
public void validarCorreoTest(String correo, boolean resultadoEsperado){
    //Arrange
    Usuario usuario = new Usuario();
    //Act Assert
    assertEquals(usuario.validarCorreo(correo), resultadoEsperado);
}
```

Figura 36

Resultados de prueba unitaria para validar correo



En la **Figura 37** se puede observar una prueba de integración para la creación de usuario, en la que se validan posibles escenarios con distintos errores; en la primera cadena de texto contiene datos correctos para todos los campos, por lo que no se debe generar ningún mensaje de error; al ser una prueba de integración, se validan varios posibles errores a en la misma prueba, como por ejemplo, que el campo nombre se deje vacío, como ocurre en el segundo escenario en el que el método mostraría el mensaje de error “El campo nombre es obligatorio”, o error en el tamaño de la contraseña, como ocurre en el último escenario, en el que el método debe mostrar el error “La contraseña debe contener al menos 8 caracteres”. En la **Figura 38** se muestra los resultados de esta prueba de integración.

Figura 37

Prueba de integración para creación de usuario

```
@Test
@Parameters({"stefany, stefany@gmail.com, 123456789, ",
            ", stefany@gmail.com, 123456789, El campo nombre es obligatorio",
            "stefany,, 123456789, El campo correo es obligatorio",
            "stefany, stefany@gmail.com,, El campo contraseña es obligatorio",
            "stefany, stefany@gmail.com, 123456789, El correo no cumple con el patrón correo@correo.com",
            "stefany, stefany@gmail, 123456789, El correo no cumple con el patrón correo@correo.com",
            "stefany, stefany@gmail.com, 123456, La contraseña debe contener al menos 8 caracteres"})
public void validarUsuarioTest(String nombre, String correo, String contrasena, String excepcion){
    //Arrange
    ExceptionInInitializerError resultado = new ExceptionInInitializerError( s: "");
    // Act
    try {
        new UsuarioBuilder()
            .UsuarioBuilderConNombre(nombre)
            .UsuarioBuilderConCorreo(correo)
            .UsuarioBuilderConContrasena(contrasena)
            .build();
    } catch (ExceptionInInitializerError e) {
        resultado = e;
    }
    // Assert
    assertEquals(resultado.getMessage(), excepcion);
}
```

Figura 38

Resultados de prueba de integración para creación de usuario

```
▼ ✓ validarUsuarioTest
  ✓ usuarioTest.validarUsuarioTest(stefany, stefany@gmail.com, 123456789, ) [0]
  ✓ usuarioTest.validarUsuarioTest(, stefany@gmail.com, 123456789, El campo nombre es obligatorio) [1]
  ✓ usuarioTest.validarUsuarioTest(stefany,, 123456789, El campo correo es obligatorio) [2]
  ✓ usuarioTest.validarUsuarioTest(stefany, stefany@gmail.com,, El campo contraseña es obligatorio) [3]
  ✓ usuarioTest.validarUsuarioTest(stefany, stefany@gmail.com, 123456789, El correo no cumple con el patrón correo@correo.com) [4]
  ✓ usuarioTest.validarUsuarioTest(stefany, stefany@gmail, 123456789, El correo no cumple con el patrón correo@correo.com) [5]
  ✓ usuarioTest.validarUsuarioTest(stefany, stefany@gmail.com, 123456, La contraseña debe contener al menos 8 caracteres) [6]
```

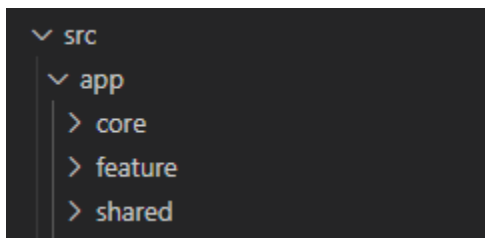
Desarrollo de plataforma administrativa

La plataforma web administrativa se desarrolló empleando el Framework Angular, por lo que el proyecto está basado en el lenguaje TypeScript y emplea también lenguaje de marcado HTML, como se explicará más adelante, por otro lado, el editor de código utilizado fue Visual Studio Code.

Arquitectura modular. Para el desarrollo de la plataforma, se ha implementado la arquitectura modular con los módulos Core – Feature – Shared, como se ve en la **Figura 39**, que permite organizar los archivos que conforman el proyecto, en el módulo correspondiente según su función.

Figura 39

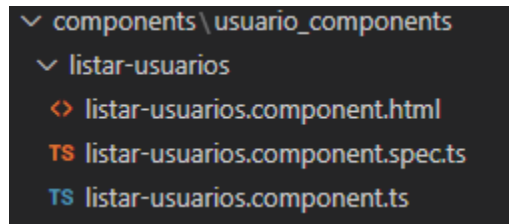
Módulos Core - Feature – Shared en Angular



Componentes. Cada funcionalidad de la plataforma web, es representada con un componente, el cual consta de tres archivos, cada uno con una función diferente, en la **Figura 40** se puede observar el componente “listar-usuarios” con sus tres archivos que tienen diferentes extensiones; por un lado, el primer archivo, con extensión .html, se encarga de la configuración de elementos visuales de la plataforma correspondientes a ese componente, y para ello se emplea, por supuesto, el lenguaje de marcado HTML; por otro lado, en el archivo con extensión .spec.ts se llevan a cabo las pruebas unitarias del componente, empleando lenguaje TypeScript; y, finalmente, el archivo con extensión .ts es quien se encarga de la lógica del componente, es decir, es el que da la funcionalidad, y emplea también el lenguaje TypeScript.

Figura 40

Componente listar-usuarios



Desarrollo de backend

En el backend del proyecto se crearon los servicios que serían consumidos desde la aplicación móvil y la plataforma web, para esto se empleó el lenguaje de programación Java, y se usó la herramienta Spring Boot, la cual está basada en Spring Framework; por otro lado, el entorno de desarrollo utilizado fue IntelliJ IDEA.

Arquitectura de microservicios. Este proyecto está basado en una arquitectura de microservicios, por lo cual, el backend está dividido en dos componentes, el primero, mostrado en la **Figura 41**, contiene todos los servicios relacionados con la gestión de usuarios y el segundo, mostrado en la **Figura 42**, contiene los servicios relacionados con las lecciones de la aplicación.

Figura 41

Componente para usuarios

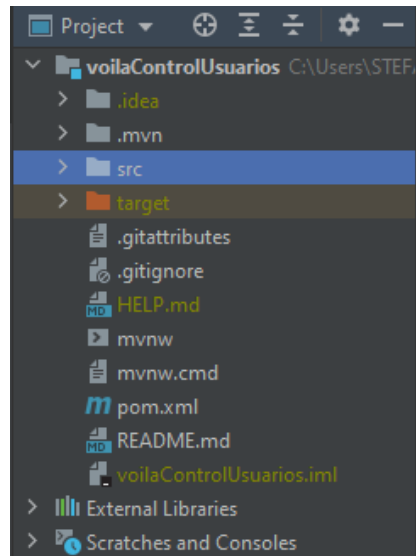
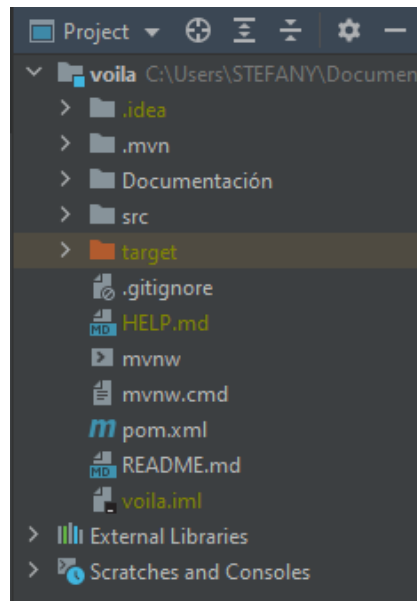


Figura 42

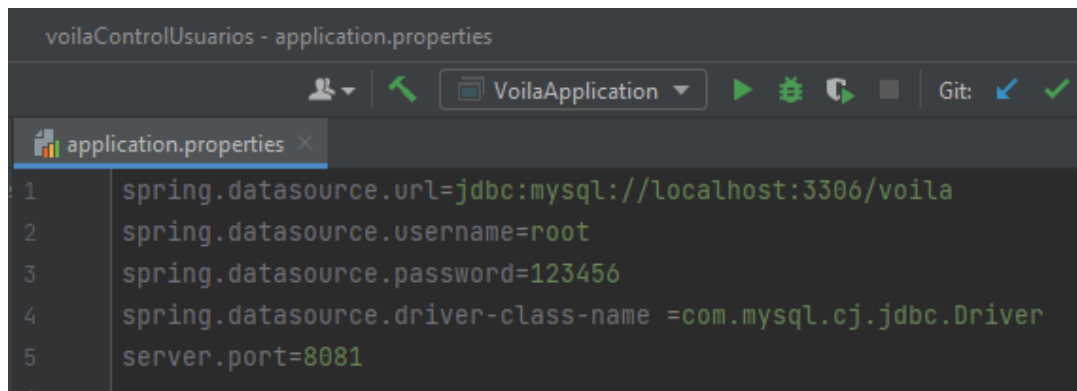
Componente para lecciones



Al usar la modularidad propia de la arquitectura de microservicios se obtienen importantes ventajas como el hecho de poder realizar modificaciones a un componente y desplegarlo mientras el otro sigue funcionando con normalidad, ya que como se puede ver en la **Figura 43** y la **Figura 44**, cada componente, funciona en un puerto distinto; por otro lado, ambos componentes se conectan a la misma base de datos como se puede observar en las figuras mencionadas anteriormente, por lo que se evidencia que, aunque los componentes estén separados, siguen siendo parte del mismo sistema.

Figura 43

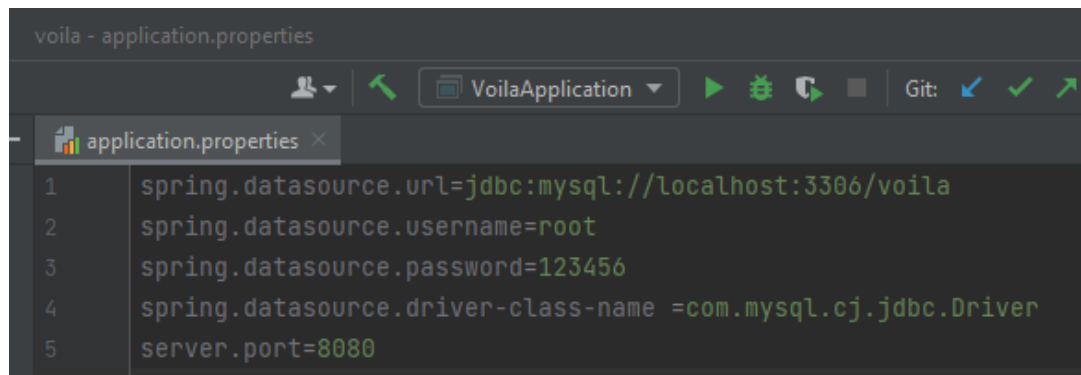
Conexión a base de datos y configuración de puerto de componente usuarios



```
voilaControlUsuarios - application.properties
VoilaApplication
application.properties
1  spring.datasource.url=jdbc:mysql://localhost:3306/voila
2  spring.datasource.username=root
3  spring.datasource.password=123456
4  spring.datasource.driver-class-name =com.mysql.cj.jdbc.Driver
5  server.port=8081
```

Figura 44

Conexión a base de datos y configuración de puerto de componente lecciones

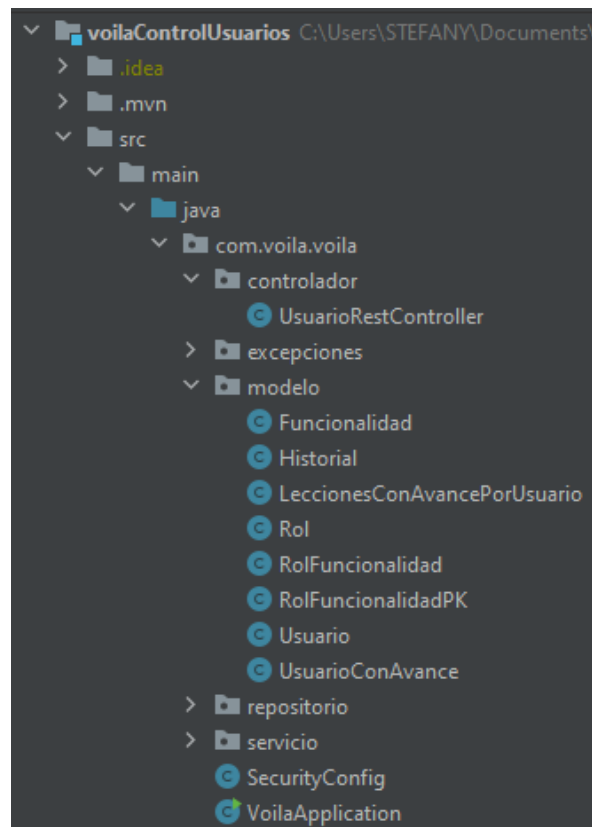


```
voila - application.properties
VoilaApplication
application.properties
1  spring.datasource.url=jdbc:mysql://localhost:3306/voila
2  spring.datasource.username=root
3  spring.datasource.password=123456
4  spring.datasource.driver-class-name =com.mysql.cj.jdbc.Driver
5  server.port=8080
```

Arquitectura MVC. Cada componente del backend sigue internamente una arquitectura MVC (Modelo – Vista – Controlador), en la **Figura 45** se muestran los paquetes pertenecientes al componente de usuarios; como se puede observar, existe un paquete “Modelo” y un paquete “Controlador”, más otros paquetes necesarios para la creación de los servicios, sin embargo, no existe un paquete “Vista”, esto se debe a que la vista corresponde a la aplicación móvil y la plataforma web, por lo que en este caso, no hay una capa de vista dentro del proyecto de Spring Boot; de manera similar se encuentra estructurado el componente de lecciones.

Figura 45

Arquitectura MVC en componente usuarios



Seguridad en el registro de usuarios. Con el fin de proporcionar seguridad en el proceso de registro de usuarios, se implementó la clase BCryptPasswordEncoder, ver **Figura 46**, perteneciente al framework Spring Security, que permite almacenar los usuarios en la base de datos con la contraseña encriptada, de manera que, si se accede a la base de datos de forma externa, un usuario registrado se ve como en la **Figura 47**.

Figura 46

Metodo de registro de usuario usando encriptación de contraseña

```
public ResponseEntity<? extends Object> registrar(Usuario usuario, BindingResult resultado) {
    Usuario usuarioValidar= new Usuario(usuario.getNombre(),usuario.getCorreo(),
        usuario.getContrasena(),usuario.getIdRol());
    usuarioValidar.setContrasena(bCryptPasswordEncoder.encode(usuarioValidar.getContrasena()));
```

Figura 47

Usuario registrado en base de datos con contraseña encriptada

	id_usuario	nombre	correo	contrasenia	id_rol
►	32	Stefany Florez	stefany@gmail.com	\$2a\$10\$ZCe42GIDNtVK3Bxt5..iNuJjB6r7.ZRCSHUYUg...	2

La misma instancia de la clase BCryptPasswordEncoder es usada para el inicio de sesión, en este proceso, la contraseña que se enviaría desde la aplicación móvil se encuentra descriptada, mientras que la contraseña almacenada en la base de datos se encuentra encriptada, por lo que, es responsabilidad del backend verificar que ambas contraseñas sean equivalentes por medio del Encoder, como se muestra en la **Figura 48**.

Figura 48

Validación de contraseña para inicio de sesión

```
return bCryptPasswordEncoder.matches (contrasena, contrasenaBD)?  
    ResponseEntity.ok().build() : ResponseEntity.badRequest().body(ERROR_INICIO_SESION);
```

Base de datos

El proyecto cuenta con una base de datos relacional, la cual fue creada empleando el sistema de gestión de bases de datos MySQL, el modelo relacional de la base de datos se puede ver en la **Figura 12** y las tablas creadas en la **Figura 49**; la inserción y actualización de datos se realiza, como se explicó en el apartado anterior, mediante el consumo de servicios que se encuentran en los componentes del backend, no obstante, debido a las necesidades del proyecto, se creó un disparador para que, al registrar un usuario, automáticamente se crearan, desde la base de datos, las diez lecciones y se establecieran en estado “sin iniciar”, este disparador se puede ver en la **Figura 50**.

Figura 49

Tablas de la base de datos en MySQL

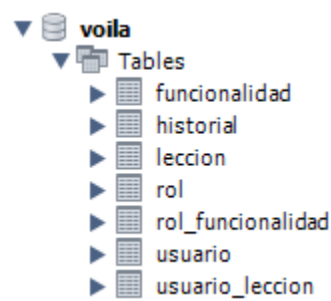


Figura 50

Disparador para creación de lecciones

```
CREATE TRIGGER insert_usuario_leccion
AFTER INSERT
ON usuario FOR EACH ROW
BEGIN
    INSERT INTO usuario_leccion(id_usuario, id_leccion, estado)
    VALUES(new.id_usuario,1,'sin iniciar'),(new.id_usuario,2,'sin iniciar'),(new.id_usuario,3,'sin iniciar'),
    (new.id_usuario,4,'sin iniciar'),(new.id_usuario,5,'sin iniciar'),(new.id_usuario,6,'sin iniciar'),
    (new.id_usuario,7,'sin iniciar'),(new.id_usuario,8,'sin iniciar'),(new.id_usuario,9,'sin iniciar'),
    (new.id_usuario,10,'sin iniciar');
END;
```

Fase de Estabilización

En esta sección se describe la manera en la que se realizó la integración de las diferentes partes de proyecto, las cuales corresponden a backend, aplicación móvil y plataforma web.

Integración de la aplicación móvil con el backend

Para el consumo de servicios desde la aplicación móvil, se empleó la librería Retrofit, con la cual se envían los datos que el servicio espera recibir en formato JSON. A continuación, se describe el proceso para el consumo de un servicio, tomando como ejemplo en este caso, el registro de un usuario.

Registro de usuarios. El proceso inicia cuando el usuario ingresa los datos en la aplicación móvil, por medio de la interfaz gráfica que se muestra en la figura **Figura 51**, allí la capa de vista captura esos datos y los envía, siguiendo el flujo de la arquitectura MVP, al presentador que lo transmite a la capa de modelo, la cual, como se evidencia en la **Figura 52**, se encarga de crear el objeto JSON a partir de los datos recibidos, y posteriormente instanciar Retrofit, haciendo llamado a la API que contiene la declaración del método para registro de usuario, con el endpoint por medio del cual se consumirá el servicio, ver **Figura 53**.

Figura 51

Formulario para registro de usuario



Formulario para registro de usuario. El formulario está superpuesto sobre una ilustración de la Torre Eiffel y edificios de París. Incluye los siguientes campos:

- Correo
- Nombre
Nombre y apellidos
- Contraseña
Al menos 8 caracteres
- Confirmar contraseña
Ingresa de nuevo la contraseña

Debajo de los campos, hay un botón azul con el texto **REGISTRARSE**. En la parte inferior de la pantalla, se ven los iconos de navegación de un sistema operativo móvil.

Figura 52

Registro de usuario empleando librería Retrofit

```
JSONObject jsonUsuario=new JSONObject();
jsonUsuario.put( name: "nombre",usuario.getNombre());
jsonUsuario.put( name: "correo",usuario.getCorreo());
jsonUsuario.put( name: "contrasena",usuario.getPwd());
jsonUsuario.put( name: "idRol", value: "2");
Call<ResponseBody> call = RetrofitClient
    .getInstance()
    .getApi()
    .registrarUsuario(jsonUsuario.toString());
```

Figura 53

Declaración de método registrarUsuario en API

```
@Headers("Content-Type: application/json")
@POST("registrarUsuario")
Call<ResponseBody> registrarUsuario(@Body String body);
```

Una vez consumido el servicio, el backend realiza los procesos correspondientes, como validar que el correo no se encuentre registrado todavía, y encriptar la contraseña para guardar en base de datos, y finalmente envía la respuesta al frontend que es recibida en el método onResponse de Retrofit, como se muestra en la **Figura 54**, en caso de haber registrado correctamente el usuario, se recibirá una respuesta exitosa, por otro lado, si por ejemplo, el correo con el que intenta registrarse ya ha sido registrado anteriormente, se recibirá una respuesta no exitosa acompañada con un mensaje de error que será mostrado en la interfaz de la **Figura 51**.

Figura 54

Obtención de respuesta en Retrofit

```
call.enqueue(new Callback<ResponseBody>() {  
    @Override  
    public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {  
  
        if (response.isSuccessful()) {  
            registroUsuarioPresenter.enviarRespuestaRegistro(REGISTRO_EXITOSO);  
        } else {  
            try {  
                RESPUESTA_SERVICIO = response.errorBody().string();  
                registroUsuarioPresenter.enviarRespuestaRegistro(RESPUESTA_SERVICIO);  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
})
```

Envío de token para inicio de sesión. Como parte de la seguridad del sistema, se ha implementado el envío de un token JWT desde la aplicación móvil que es recibido y validado en el backend, como método de autenticación de los usuarios, en la **Figura 55** se puede ver el método usando en Android Studio para generar el token, al cual se le establece una fecha de expiración de un día, y en la **Figura 56**, se ve cómo el token generado es enviado en el consumo de servicio mediante Retrofit.

Figura 55

Método para generar token en Android Studio

```
private String getJWTToken(String username) {  
    String key = username;  
    Algorithm algoritmo = Algorithm.HMAC256(key);  
    Date currentTime = new Date();  
    String token = JWT.create()  
        .withIssuer("voila")  
        .withAudience("frontend")  
        .withExpiresAt(new Date(currentTime.getTime() + 24*3600*1000L))  
        .sign(algoritmo);  
  
    return token;  
}
```

Figura 56

Consumo de servicio para inicio de sesión enviando token generado

```
String token = getJWTToken(correo);  
  
Call<ResponseBody> call = RetrofitClient  
    .getInstance()  
    .getApi()  
    .iniciarSesion(correo,contrasena,token);
```

Una vez recibidos los datos de inicio de sesión en el backend, se realizan las validaciones necesarias para permitir o denegar el acceso; en la **Figura 57**, se puede ver que se hace un llamado al método de verificar token, mostrado en la **Figura 58**, y de obtenerse un resultado verdadero o “true” se procede a validar que la contraseña ingresada sea correcta, pero si al verificar el token, se obtiene un resultado falso o “false”, el acceso es denegado y se envía como respuesta del servicio un mensaje de error.

Figura 57

Validaciones en Spring Boot para inicio de sesión

```
if(verificarToken(correo,token)){
    return bCryptPasswordEncoder.matches (contrasena, contrasenaBD)?
        ResponseEntity.ok().build() : ResponseEntity.badRequest().body(ERROR_INICIO_SESION);
}else {
    return ResponseEntity.badRequest().body(ERROR_INICIO_SESION);
}
```

Figura 58

Método para verificación de token en Spring Boot

```
private boolean verificarToken(String username, String token){
    String key = username;
    Algorithm algoritmo = Algorithm.HMAC256(key);
    JWTVerifier verifier = JWT.require(algoritmo)
        .withIssuer("voila")
        .withAudience("frontend")
        .build();
    try{
        verifier.verify(token);
    } catch (JWTVerificationException e) {
        e.printStackTrace();
        return false;
    }
    return true;
}
```

Integración de plataforma web con el backend

Para el consumo de servicios desde la plataforma web, se empleó la librería de `HttpService`, con la cual se obtiene la información en formato JSON; a continuación, se describe el proceso para consumir un servicio tomando como ejemplo el servicio para listar usuarios con avances.

Listar usuarios con avance. El proceso comienza cuando el usuario administrador accede al enlace de Listar usuarios con avance que se encuentra resaltado en la **Figura 59**, lo que hace que se cargue el componente correspondiente, perteneciente al módulo Feature, este componente realiza un llamado al archivo de servicio del módulo Shared, como se puede observar en la **Figura 61**, el cual, por medio de la librería HttpService (ver figura **Figura 60**), consume el servicio que obtiene en formato JSON la información solicitada, para posteriormente devolverla al componente, donde es asignada a una lista del modelo como se puede ver en la **Figura 61**, de la cual se obtiene la información para mostrar en la vista, por medio de HTML como se muestra en la **Figura 62**.

Figura 59

Vista de la plataforma con los resultados de usuarios con avance

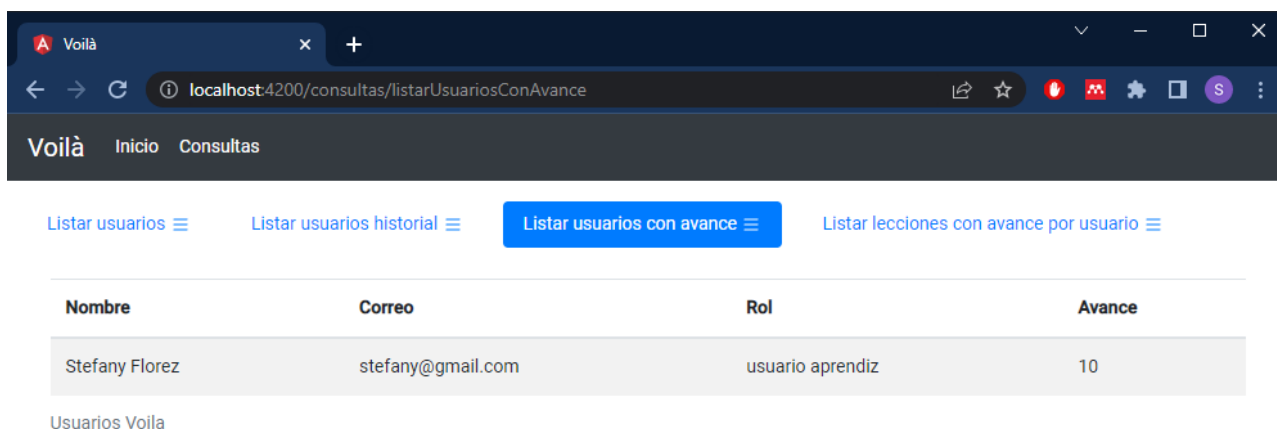


Figura 60

Consumo de servicio para la consulta de usuarios con avance

```
public consultarUsuariosConAvance() {  
    return this.http.doGet<Usuario[]>('http://localhost:8081/voila/usuario/consultarUsuariosConAvance',  
    this.http.optsName('consultar usuarios con avance'));  
}
```

Figura 61

Asignación de los datos obtenidos en el consumo del servicio para consulta de usuarios con avance

```
ngOnInit() {  
  this.listaUsuarios = this.usuarioService.consultarUsuariosConAvance();  
}
```

Figura 62

Estructura HTML para mostrar los datos obtenidos

```
<tbody>  
  <tr id="trListar" *ngFor="let usuario of listaUsuarios | async">  
    <td>{{usuario.nombre}}</td>  
    <td>{{usuario.correo}}</td>  
    <td>{{usuario.rolUsuario}}</td>  
    <td>{{usuario.avance}}</td>  
  </tr>  
</tbody>
```

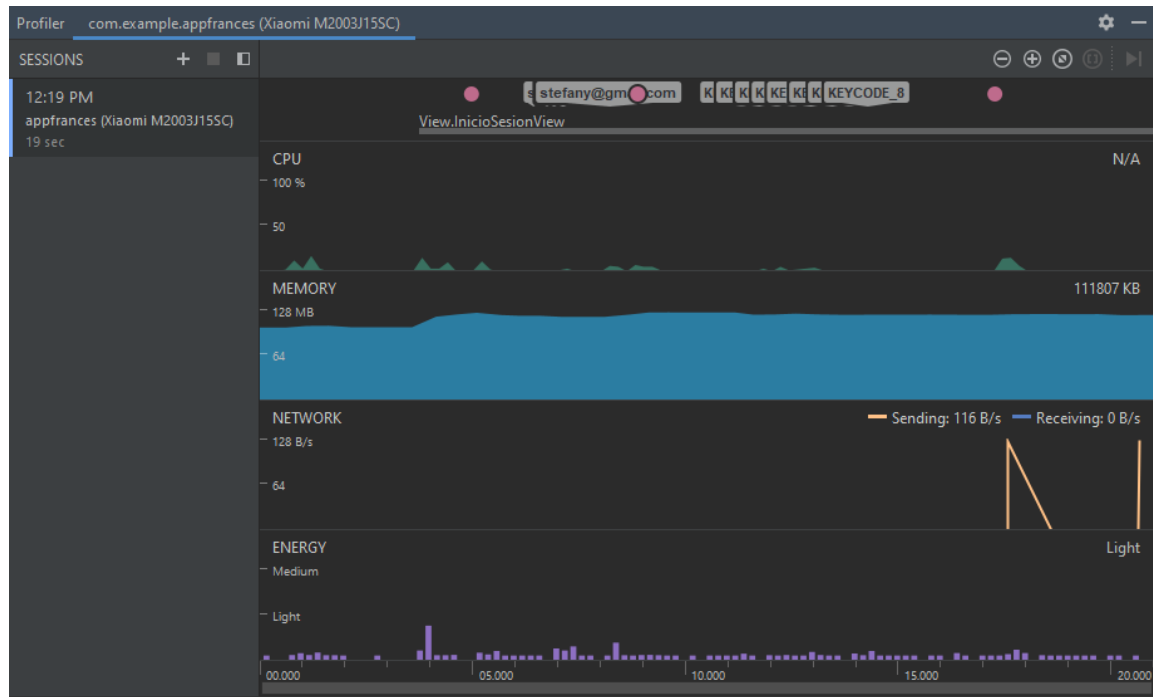
Fase de Pruebas del Sistema

Pruebas de rendimiento

Para medir el rendimiento de la aplicación móvil, se usó la herramienta de Android Studio llamada Android Profiler, que permite visualizar en tiempo real el uso de CPU, memoria, red y energía, en el dispositivo móvil; la **Figura 63** muestra la medición del uso de estos recursos durante el inicio de sesión.

Figura 63

Medición de uso de recursos en dispositivo Xiaomi durante el inicio de sesión



Como se puede observar en la **Figura 63**, hay dos puntos en la línea de tiempo donde la medición varía notablemente; el primero, en el momento en que se despliega la vista de inicio de sesión, esto ocurre alrededor del segundo 3; en la **Figura 64**, se ve que el uso de CPU en ese momento es de 3%, y en la **Figura 65**, se ve el uso de memoria al haber entrado a la vista, que es de 103,9 MB, también se puede notar que hay un aumento en el consumo de energía en ese momento, sin embargo, al igual que durante toda la medición, el consumo es bajo.

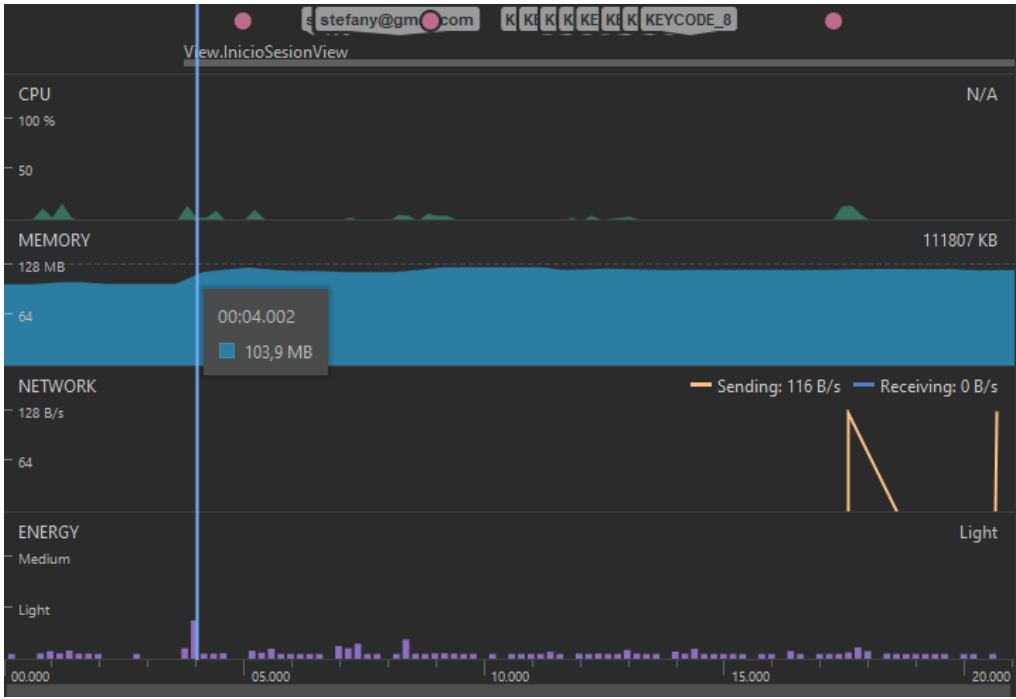
Figura 64

Uso de CPU al entrar a la vista de inicio de sesión



Figura 65

Uso de memoria al entrar a la vista de inicio de sesión



El segundo momento en que se observa notable variación en las mediciones, a excepción de la memoria, es al presionar el botón de iniciar sesión; en la **Figura 66**, se muestra el uso de CPU de 15%, que es el más alto presentado durante todo el tiempo de medición; en la **Figura 67**, se muestra un uso de memoria de 110,1 MB, el cuál a diferencia de los otros recursos, no muestra variación considerable en este punto; por otro lado, el uso de red presenta medición por primera vez, como se puede observar en la **Figura 68**.

Figura 66

Medición de uso de CPU al presionar el botón INICIAR SESIÓN



Figura 67

Medición del uso de memoria al presionar el botón INICIAR SESIÓN

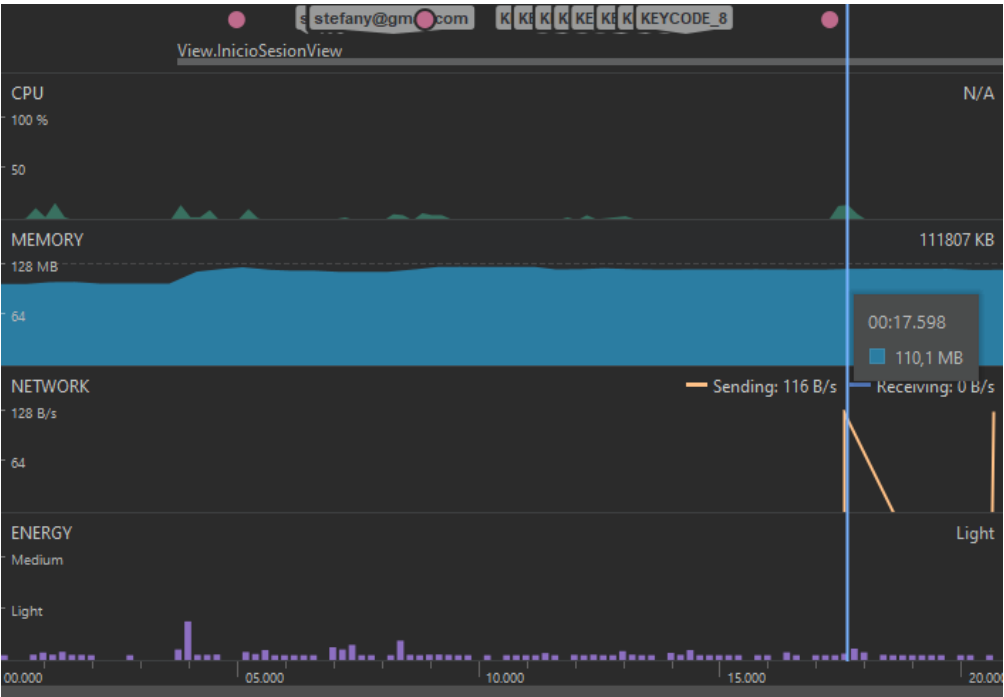
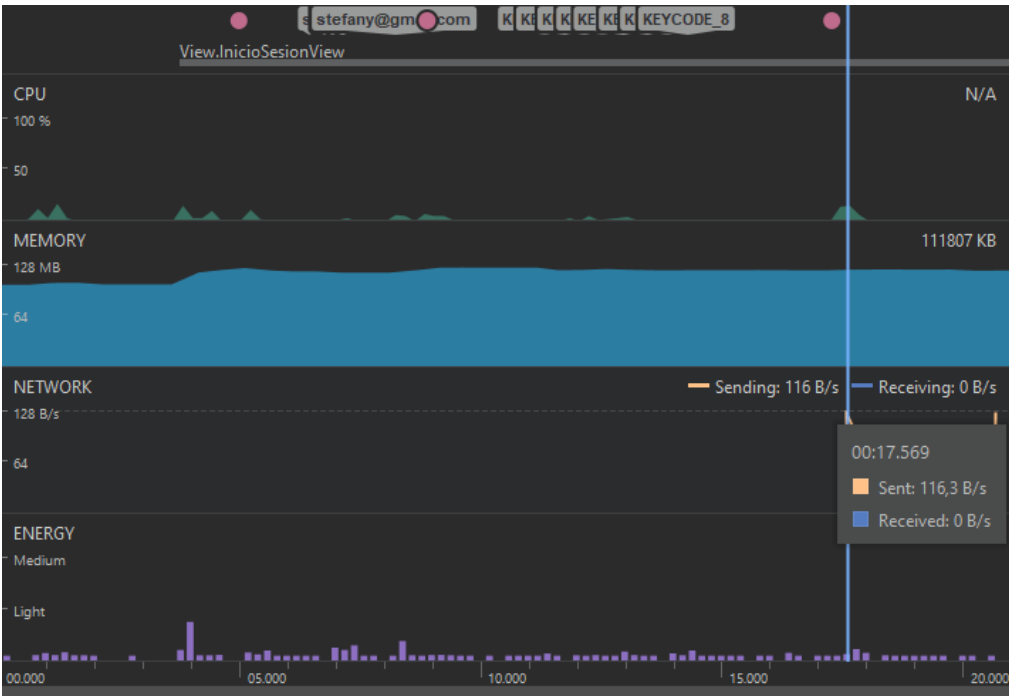


Figura 68

Medición del uso de red al presionar el botón INICIAR SESIÓN



Se realizaron pruebas de carga para validar el rendimiento de los servicios bajo condiciones de alta demanda, para esto se utilizó el software Apache JMeter; a continuación, se muestra como ejemplo las pruebas al servicio de consultar usuarios, para lo cual se simulaban las peticiones de 1000 usuarios en un tiempo de 60 segundos, como se muestra en la **Figura 69**.

Configuración de número de peticiones y tiempo

Thread Properties

Number of Threads (users): 1000

Ramp-up period (seconds): 60

Al finalizar las 1000 peticiones se obtuvieron los resultados mostrados en la **Figura 70**, donde se comprueba que el servicio respondió correctamente a la cantidad de peticiones realizadas en el tiempo establecido, pues se obtuvo un error de 0%, con un rendimiento de 16.7 peticiones por segundo.

Resultados de la prueba de carga para consulta de usuarios

[illegible]

Conclusiones

Analizar los beneficios que proporciona una herramienta de aprendizaje, en comparación con otra, contribuye a optimizar el aprovechamiento de las mismas dado que permite seleccionar adecuadamente la herramienta que mejor se adapte a las necesidades del proceso de enseñanza-aprendizaje para un área educativa específica.

Realizar correctamente las etapas de diseño y planificación de un proyecto de software, proporciona una base sólida que permite llevar a cabo el desarrollo del proyecto de forma eficiente, aprovechando los recursos de tiempo y personal, ya que se genera un plan de acción orientado al cumplimiento de los requisitos establecidos inicialmente, que finalmente lleva a la obtención de un producto estable y funcional.

El uso de gamificación en un entorno de aprendizaje aporta importantes ventajas para la adquisición de conocimiento, ya que permite, mediante las sensaciones generadas durante el juego, brindar una experiencia agradable al usuario, que lo impulse a prestar especial atención a la información que se le proporciona y lo motive a incrementar el tiempo de uso, en este caso, de la aplicación.

La implementación de una arquitectura que proporcione modularidad, como la arquitectura de microservicios, puede ser beneficioso en gran medida para el desarrollo de un software, ya que actualmente esta es un área que está en constante crecimiento, por lo que el software se debe construir pensando en la escalabilidad, y a medida que este crece, se va haciendo necesario, aún más, tener los diferentes componentes de forma separada, ya que se debe ser capaz de realizar ajustes a una funcionalidad, sin tener que detener el sistema completo, lo que es posible gracias a la modularidad.

Referencias

- Álvarez Casadiego, C. A. (2018). Estudio del uso de la metodología ágil SCRUM en el desarrollo de una aplicación nativa para la plataforma Android en un entorno con requisitos cambiantes caso de aplicación app compositores.
- Amaya Balaguera, Y. D., (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. Revista de tecnología, Vol 12,111-124.
- Arango Giraldo, S. (2018). Estudio de factibilidad de la creación de una aplicación digital, que permita la enseñanza del idioma inglés a niños con dislexia, bajo los lineamientos de la metodología ONUDI.
- Arroyo, N. (2011). Información en el móvil. Barcelona: UOC.
- Figueroa Hernandez, A. M., (2019).Desarrollo de aplicación móvil (app) para la inclusión de estudiantes con discapacidad diferencial auditiva. Caso de estudio: asignatura Multimedia y Laboratotio del programa de Ingeniería en Telecomunicaciones de la Universidad de Pamplona sede Villa del Rosario.
- Freepik company. (s.f.). *Flaticon*. Obtenido de <https://www.flaticon.com/>
- Gómez Ardila, Sergio Eduardo, Chacón González, Ana Carolina Aprendizaje móvil basado en el modelo Frame y aplicado al aprendizaje de la técnica de Core en Fisioterapia. Revista Virtual Universidad Católica del Norte [en línea]. 2017, (50), 411-436[fecha de Consulta 9 de Agosto de 2021]. ISSN: 0124-5821. Disponible en: <https://www.redalyc.org/articulo.oa?id=194250865023>
- Grinder, J., & Bandler, R. (2007). *De sapos a príncipes*. Santiago: Cuatro vientos.
- Gualancañay Tomalá, N. P. (2018). Estilos de aprendizaje en los estudiantes de bachillerato. *Dominio*

de las ciencias, Vol 5, 488-504.

<http://dx.doi.org/10.23857/dom.cien.pocaip.2019.5.1.enero.488-504>

Guy, R. (2009). *The Evolution of Mobile Teaching and Learning*. Santa Rosa: Informing Science Press.

Haro, E., Guarda, T., Zambrano Peñaherrera¹, A. O., Ninahualpa Quiña¹, G., (2019). Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot. *Revista Ibérica de Sistemas e Tecnologías de Informação Vol 17*, 309-321.

Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. San Francisco: Pfeiffer.

Krashen, S. D. (1981). *Second Language Acquisition and Second Language Learning*. Oxford: Pergamon Press Inc.

Krashen, S. D. (1985). *The Input Hypothesis. Issues and Implications*. London: Longman Group UK Ltd.

Kumar Basak, S., Wotto, M., & Bélanger, P. (2018). E-learning, M-learning and D-learning: Conceptual definition and comparative analysis. *e-learning and digital media*, 191-216.

Lopez, D., Maya, E., (2017). Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web. *Séptima Conferencia de Directores de Tecnología de Información, TICAL 2017 Gestión de las TICs para la Investigación y la Colaboración, San José, del XX al XX de julio de 2017*.

LottieFiles. (s.f.). *LottieFiles*. Obtenido de <https://lottiefiles.com/>

Moreno, A. I., & Vermeulen, A. (2015). Using VISP (VIdeos for SPeaking), a Mobile App Based on Audio Description, to Promote English Language Learning among Spanish Students: A Case Study. *Procedia - Social and Behavioral Sciences*, 178, 132-138.

<https://doi.org/10.1016/j.sbspro.2015.03.169>

Nosseir, A., Flood, D., Harrison, R., & Ibrahim, O. (2012). Mobile Development Process Spiral.

International Conference on Computer Engeneering and Systems, (págs. 281-286).

Observatorio de Innovación Educativa del Tecnológico de Monterrey. (Septiembre de 2016).

Gamificación en la educación. Obtenido de eduteka:

<http://eduteka.icesi.edu.co/articulos/edutrends-gamificacion>

Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (2013). Directrices

de la UNESCO para las políticas de aprendizaje móvil. Recuperado de

<https://unesdoc.unesco.org/ark:/48223/pf0000219662>

Ortiz Colón, A. M., Jordan, J., Agredal, M., (2018). Gamificación en educación: una panorámica

sobre el estado de la cuestión. *Educação e Pesquisa*, vol. 44, e173773. DOI: 10.1590/S1678-

4634201844173773

Pegalajar Palomino, M.C. (2021). Implicaciones de la gamificación en Educación Superior: una

revisión sistemática sobre la percepción del estudiante. *Revista de Investigación Educativa*,

39(1), 169-188. DOI: <http://dx.doi.org/10.6018/rie.419481>

Pontes, H. P., Furlan Duarte, J. B., & Pinheiro, P. R. (2020). An educational game to teach numbers

in Brazilian Sign Language while having fun. *Computers in Human Behavior*, 107, 105825.

<https://doi.org/10.1016/j.chb.2018.12.003>

Quevedo Seminario, S., Díaz Alarcón, M., & La Rosa Feijoo, O. C. (2021). Método auditivo visual

para el aprendizaje del idioma inglés en los estudiantes de turismo. *Cuadernos de desarrollo*

aplicados a las TIC, 93-120.

Roldan Martinez, D., Valderas Aranda, P. J., Torres Bosch, V. (2018). Microservicios, un enfoque

integrado. Madrid. RA-MA Editorial.

- Sahani, A. K., Singh, P. (2020) Web Development Using Angular: A Case Study. *Journal of Informatics Electrical and Electronics Engineering*, 1(2), 5, 1-7.
- Sanchez Acosta, W. G. (). Desarrollo de una aplicación de realidad aumentada para dispositivos móviles Android, que refuerce el aprendizaje del idioma inglés nivel A1 en niños de 10 a 12 años de edad.
- Sommerville, I. (2011). *Ingeniería de Software*. México. Pearson Educación.
- Stieglitz, S., Latteman, C., Robra-Bissantz S., Zarnekow, R., Brockmann T. (2017). *n niños Gamification. Using game elements in serious contexts*. Cham: Springer International Publishing.
- Trillos Arenas, J., Hernandez Villamizar, D. A., Barrientos Avendaño, E., Rico Bautista, D., Coronel Rojas, L., Cuesta Quintero, F. (2019). Proyecto caprino de la Universidad Francisco de Paula Santander Ocaña: Caso de aplicación de tecnología. *Revista Ibérica de Sistemas y Tecnologías de la Información*, 23, 94-106.
- Wu, Q. (2015). Designing a smartphone app to teach English (L2) vocabulary. *Computers and Education*, 85, 170-179. <https://doi.org/10.1016/j.compedu.2015.02.013>