

UNIVERSIDAD UNIFRANZ

FACULTAD DE INGENIERÍA



CARRERA INGENIERÍA DE SISTEMA

PERFIL DE PROYECTO

TEMA: SISTEMA DE MICROSERVICIOS PARA

LA PLAZA DE COMIDAS LINCOLN

ASIGNATURA:

Programación de Dispositivos Móviles

NOMBRE:

VILLARROEL VALENCIA ANDRÉ NICOLÁS

DOCENTE:

Ing. Yony Richard Montoya Burgos

CBBA-BOLIVIA

2024

ÍNDICE GENERAL

INTRODUCCIÓN	2
CAPÍTULO I: ASPECTOS GENERALES	2
1.1. Descripción del problema	2
1.2. Objetivos	2
1.2.1. Objetivos generales	2
1.2.2. Objetivos específicos	2
1.3. Alcances y limitaciones	3
1.3.1. Alcance	3
1.3.2. Límites	3
CAPÍTULO II: MARCO TEÓRICO	3
2.1. Aplicación móvil	3
2.2. Herramientas de desarrollo	4
2.2.1. JavaScript	4
2.2.2. JSX	5
2.2.3. CSS	6
2.2.5. React Native	6
2.2.6. Expo	7
2.3. Sistema operativo Android	8
CAPÍTULO III: DISEÑO DEL SISTEMA	9
3.1. Diseño de Figma	9
3.2. Vistas del sistema	12

INTRODUCCIÓN

El presente proyecto trata sobre el desarrollo de una aplicación móvil donde permite crear y compartir una lista de deseos o regalos que te gustaría recibir. La idea principal es facilitar la organización de regalos entre amigos y familiares, evitando las compras duplicadas y asegurando que recibas lo que realmente deseas.

Esta aplicación se está desarrollando con el framework de React Native que es mucho más ligero y no consume muchos recursos de las computadoras como otras herramientas de desarrollo móvil

CAPÍTULO I: ASPECTOS GENERALES

1.1. Descripción del problema

Dificultad para organizar regalos entre amigos y familiares de manera eficiente y sin duplicaciones, especialmente en eventos como cumpleaños, fiestas o celebraciones especiales.

1.2. Objetivos

1.2.1. Objetivos generales

Desarrollar una aplicación móvil según las indicaciones en clases, con el lenguaje o herramientas más convenientes para su desarrollo.

1.2.2. Objetivos específicos

- Diseñar una interfaz cómoda para los usuarios
- Desarrollar las vistas requeridas
- Mostrar los regalos y sus descripciones
- Tener una conexión con la base de datos

1.3. Alcances y limitaciones

1.3.1. Alcance

- La aplicación tendrá un diseño intuitivo, se implementará una paleta de colores y un estilo visual coherente con la marca de la aplicación.
- Se utilizará React Native como framework de desarrollo.
- La aplicación abarcará las funcionalidades descritas en los objetivos específicos.

1.3.2. Límites

- Está desarrollado solo para Android
- No tendrá una personalización avanzada de los regalos

CAPÍTULO II: MARCO TEÓRICO

2.1. Aplicación móvil

Una aplicación móvil, más comúnmente conocida como app, es un tipo de software desarrollado específicamente para su uso en pequeños dispositivos informáticos inalámbricos, como smartphones y tablets, en lugar de ordenadores de sobremesa o portátiles.

Las aplicaciones nativas se desarrollan expresamente para el sistema operativo (SO) de un dispositivo móvil. Por lo tanto, existen aplicaciones móviles nativas para Android y aplicaciones móviles nativas para iOS, por no hablar de las aplicaciones para cualquier otra plataforma o dispositivo. Como están diseñadas para una plataforma específica, no puedes utilizar aplicaciones Blackberry en un teléfono Android o aplicaciones iOS en un teléfono Windows, por ejemplo. Para desarrollar aplicaciones nativas se utilizan numerosos lenguajes

de programación. Algunos ejemplos son Javascript, Kotlin, Python, Swift, Objective-C, C++ y React.

Las aplicaciones móviles están diseñadas para funcionar en sistemas operativos específicos, como iOS, Android y Windows Phone. Cuando una aplicación móvil se descarga e instala en un dispositivo, se almacena en su memoria y se ejecuta con el sistema operativo del dispositivo.

El uso de aplicaciones móviles tiene numerosas ventajas:

1. **Comodidad.** Las aplicaciones móviles pueden descargarse e instalarse en un dispositivo, lo que permite a los usuarios acceder a las funciones y servicios de la aplicación en cualquier momento y desde cualquier lugar.
2. **Personalización.** Las aplicaciones móviles pueden adaptarse a las necesidades específicas de cada usuario, proporcionando una experiencia personalizada.
3. **Acceso sin conexión.** Muchas aplicaciones móviles pueden utilizarse sin conexión, lo que permite acceder a información y funciones importantes incluso cuando no se dispone de conexión a Internet.
4. **Notificaciones push.** Las aplicaciones móviles pueden enviar notificaciones push a los usuarios, proporcionando actualizaciones en tiempo real sobre información y eventos importantes.

2.2. Herramientas de desarrollo

2.2.1. JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas

actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado.

El núcleo del lenguaje JavaScript de lado del cliente consta de algunas características de programación comunes que te permiten hacer cosas como:

- Almacenar valores útiles dentro de variables. En el ejemplo anterior, por ejemplo, pedimos que ingreses un nuevo nombre y luego almacenamos ese nombre en una variable llamada `name`.
- Operaciones sobre fragmentos de texto (conocidas como "cadenas" (strings) en programación). En el ejemplo anterior, tomamos la cadena `"Player1:"` y la unimos a la variable `name` para crear la etiqueta de texto completa, p. ej. `"Player1: Chris"`.
- Y ejecuta código en respuesta a ciertos eventos que ocurren en una página web. Usamos un evento click en nuestro ejemplo anterior para detectar cuándo se hace clic en el botón y luego ejecutar el código que actualiza la etiqueta de texto.

2.2.2. JSX

Se llama JSX, y es una extensión de la sintaxis de JavaScript. Recomendamos usarlo con React para describir cómo debería ser la interfaz de usuario. JSX puede recordarte a un lenguaje de plantillas, pero viene con todo el poder de JavaScript. JSX produce “elementos” de React. Exploraremos como renderizarlos en el DOM.

Esta sintaxis fue creada principalmente, bajo el concepto de que los componentes en React son pedazos de lógicas aisladas que hacen una función, pero se pueden compartir información entre ellas, teniendo esto en cuenta, con esta sintaxis podemos crear lógica JS, HTML y CSS en un mismo componente sin la necesidad de tenerlos en archivos aislados.

JSX se utiliza dentro del componente donde se declaran los componentes a renderizar, para esto se debe escribir cualquier expresión javascript dentro de llaves y esta podrá ser utilizada.

2.2.3. CSS

CSS, al igual que HTML, es uno de los lenguajes centrales de Internet. Mientras que para añadir texto a un sitio web se utiliza HTML y se estructura semánticamente, para definir el diseño del contenido se utiliza CSS. Aunque HTML y CSS se utilizan en combinación, las instrucciones de diseño de CSS y los elementos de HTML existen por separado. Esto significa que una máquina puede leer un documento electrónico incluso sin CSS. Con la ayuda de CSS, el contenido del navegador se prepara visualmente y se presenta de forma atractiva.

El éxito de un sitio web no depende solo del contenido, sino también de un buen diseño. Los usuarios pierden rápidamente el interés por las páginas web que no son fáciles de usar o no están bien estructuradas. En este caso, CSS ofrece una serie de opciones de diseño que no están disponibles en HTML puro.

2.2.5. React Native

React Native es un framework JavaScript para crear aplicaciones reales nativas para iOS y Android, basado en la librería de JavaScript React para la creación de componentes visuales, cambiando el propósito de los mismos para, en lugar de ser ejecutados en navegador, correr directamente sobre las plataformas móviles nativas, en este caso iOS y Android. Es decir, en lugar de desarrollar una aplicación web híbrida o en HTML5, lo que obtienes al final como resultado es una aplicación real nativa, indistinguible de la que podrías desarrollar con tu código en Objective-C o Java.

React Native usa el mismo paradigma fundamental de construcción de bloques de UI (componentes visuales con los que interacciona el usuario) que las aplicaciones nativas reales de Android e iOS, pero gestiona la interacción entre los mismos utilizando las capacidades de JavaScript y React.

En React existe un “VirtualDOM”, en el que tenemos nuestro JSX, en el cual definimos los documentos HTML, y estos se transforman en componentes del navegador a través de JavaScript. Con React Native ocurre algo parecido, ya que tenemos nuestros componentes JSX, que van a ser distintos a los componentes HTML y que tendrán otros tags y otros nombres, ya que no estamos utilizando HTML.

Lo que va a suceder es que el compilador que tiene React Native los va a convertir en elementos nativos de la interfaz para Android y para iOS, lo cual va a permitir que estas aplicaciones tengan un look and feel parecido a aplicaciones nativas, un rendimiento prácticamente igual y una experiencia de navegación y de usuario muy similar a las aplicaciones nativas, ya que lo que se está generando es interfaz nativa.

2.2.6. Expo

Expo es un framework para aplicaciones React universales. Se trata de un conjunto de herramientas y servicios construidos alrededor de React Native y de las plataformas móviles que ayudan a desarrollar, construir y desplegar aplicaciones para iOS, Android, pero también aplicaciones Web, utilizando en todos los casos la misma base de código JavaScript/TypeScript con React.

Expo proporciona casi todas las funciones que pueda imaginar usar en su aplicación. Proporciona componentes y API para casi todas las funciones nativas del dispositivo, como reproducción de video, sensores, ubicación, seguridad, información del dispositivo y mucho

más. Piense en Expo como un paquete de servicio completo que hace su vida como desarrollador de React Native mucho más fácil.

También utiliza una versión de alguna manera modificada de React Native, que normalmente está una o dos versiones por detrás de la última versión de React Native. Por lo tanto, cuando trabaja con Expo, debe esperar las últimas funciones de React Native un par de meses después de su lanzamiento. Recomendaría usar Expo si desea lograr resultados a la máxima velocidad sin tener que optimizar el tamaño de su package.

Al configurar un nuevo proyecto con Expo, puede elegir entre dos tipos diferentes de flujos de trabajo: un flujo de trabajo simple y un flujo de trabajo administrado. En ambos flujos de trabajo, el marco le proporciona bibliotecas fáciles de usar para incluir elementos nativos como la cámara, el sistema de archivos y otros.

2.3. Sistema operativo Android

El sistema operativo Android es una plataforma tecnológica diseñada para dispositivos móviles como teléfonos inteligentes, tabletas y relojes inteligentes. Desarrollado por Google, Android ofrece un entorno operativo versátil y altamente personalizable que permite a los usuarios acceder a una amplia gama de funciones y aplicaciones. Desde su lanzamiento inicial en 2008, Android se ha convertido en uno de los sistemas operativos más utilizados en el mundo, potenciando millones de dispositivos móviles a nivel global.

Desarrollado por Google, diseñado principalmente para dispositivos móviles como teléfonos inteligentes y tabletas. Se basa en el núcleo del sistema operativo Linux y ha experimentado un gran éxito en el mercado debido a su flexibilidad, personalización y gran cantidad de aplicaciones disponibles en Google Play Store.

CAPÍTULO III: DISEÑO DEL SISTEMA

3.1. Diseño de Figma

Estos diagramas fueron diseñados según la tarea dada en clases

Imagen 1: Vistas del inicio y registro

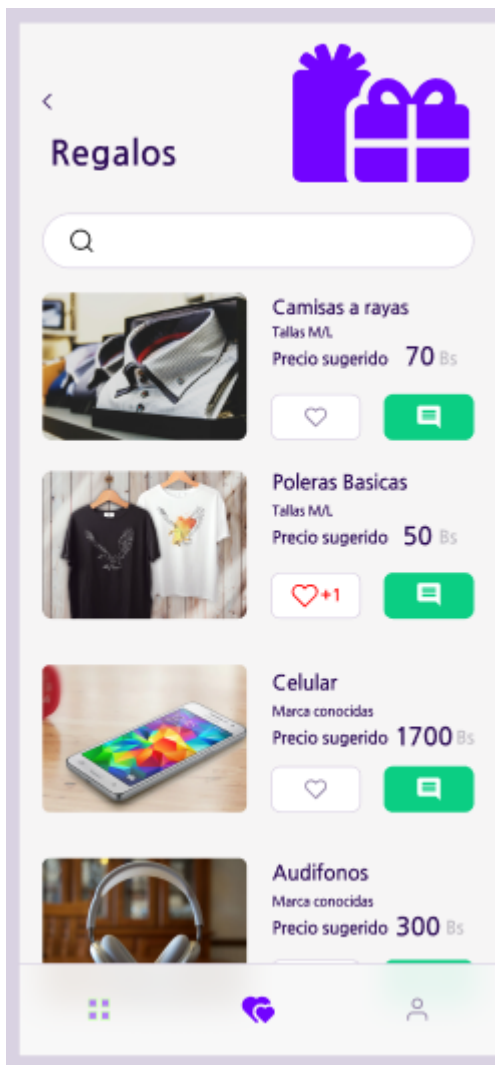


Fuente: Elaboración Propia

Imágenes 2: Vistas de Sesión y de Inicio

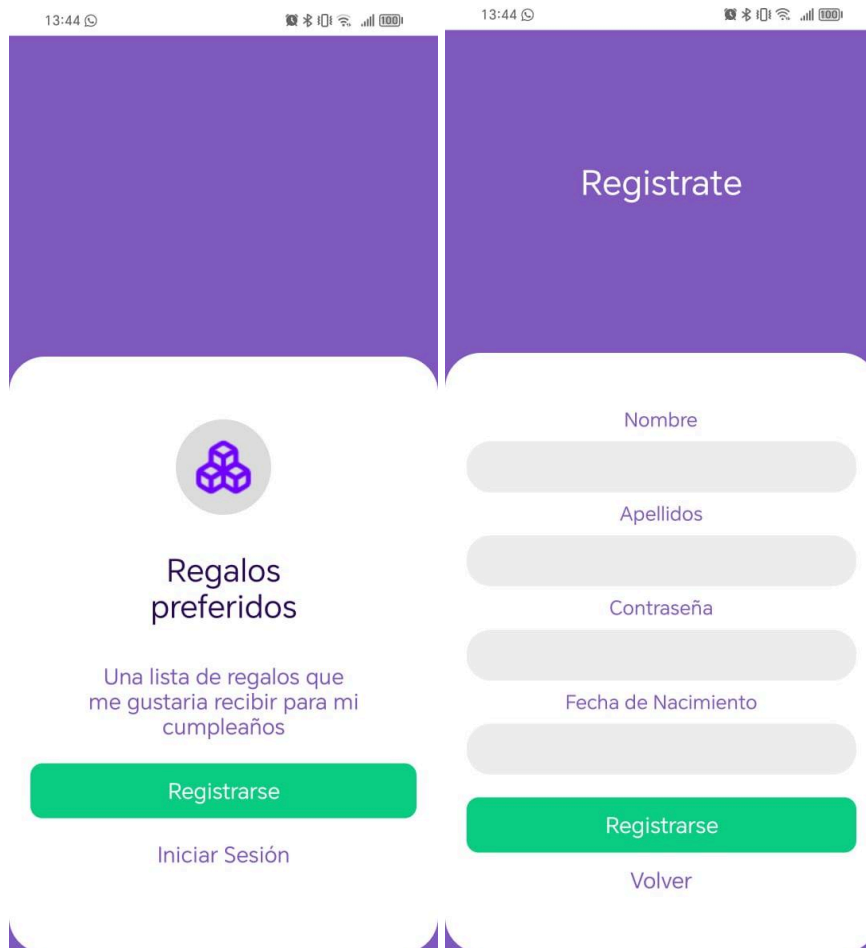


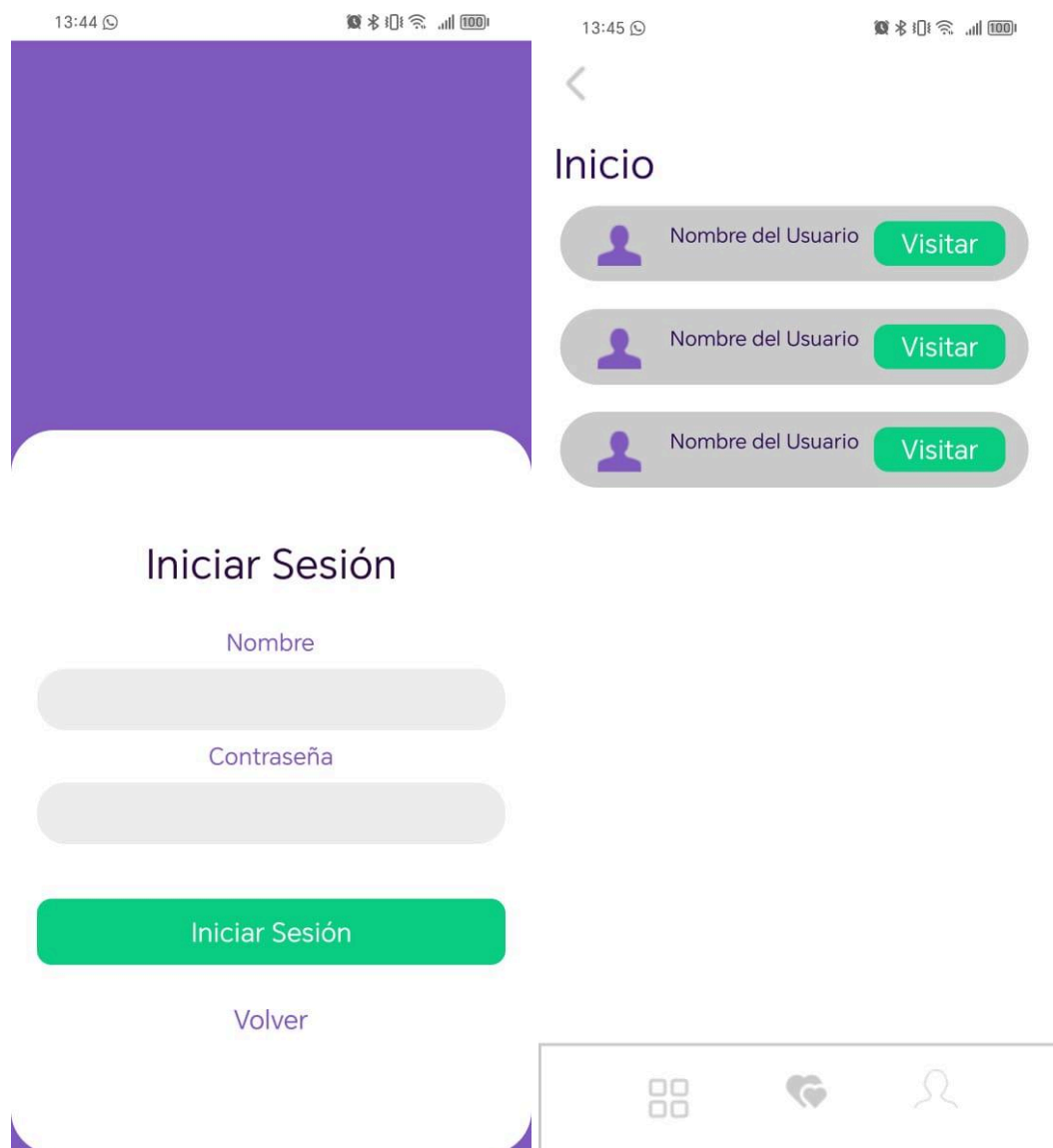
Fuente: Elaboración Propia

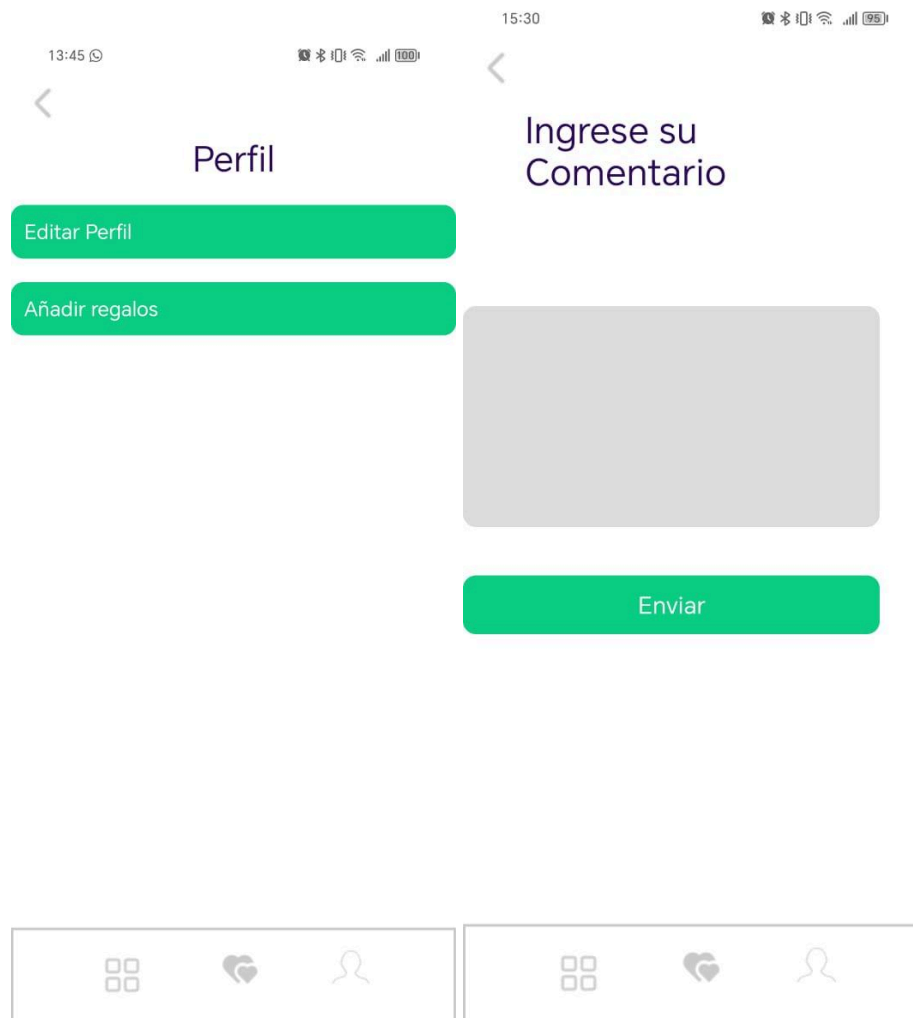


3.2. Vistas del sistema

Se demostrará el avance del proyecto mediante imágenes de las vistas desarrolladas.







15:45



Regalos



Camisas a Rayas

Talla M/L

Precio Sugerido

70 Bs



Poleras Basicas

Talla M/L

Precio Sugerido

50 Bs

