TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN
División de Estudios Profesionales

Trabajo de Titulación

Opción I: Tesis.

Proyecto: “ANÁLISIS DE HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES MÓVILES HIBRIDAS EN EL CONTEXTO DE UN SISTEMA DE TIEMPO REAL EN EL SEGUIMIENTO DE CONTAMINANTES CRITERIO.
”

ALUMNO(S):	Jonathan Alejandro Flores Saldaña
No. CONTROL:	16480879
CARRERA:	Ingeniería en Sistemas Computacionales
ASESOR DE TESIS:	Dr. José Isidro Hernández Vega
REVISORES:	

Guadalupe, N.L. JUNIO, 2021

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLOGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Nuevo León

Aceptación de documento de Tesis

Cd. Guadalupe, Nuevo León, [redacted]

ING. MAGALY BENÍTEZ TAMEZ
JEFA DE DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
PRESENTE:

La Comisión de Revisión de Tesis nos es grato comunicarle que, conforme a los lineamientos de los planes de estudio del 2010 del Tecnológico Nacional de México para la obtención del grado de Ingeniería en Sistemas Computacionales de este Instituto, y después de haber sometido a revisión académica el proyecto de Tesis titulado:**"Análisis de herramientas para el desarrollo de aplicaciones móviles hibridas en el contexto de un sistema de tiempo real en el seguimiento de contaminantes criterio"**, realizado por **Jonathan Alejandro Flores Saldaña**, Número de Control: **16480879**, dirigida por el **Dr. José Isidro Hernández Vega** y como co-directora la **M.C. Elda Reyes Varela**, y habiendo realizado las correcciones que le fueron indicadas, acordamos **ACEPTAR** el documento final de proyecto de Tesis. Así mismo le solicitamos tenga a bien extender la documentación correspondiente para continuar el proceso de titulación integral por tesis del sustentante.

Sin otro particular, agradecemos la atención.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"CIENCIA Y TECNOLOGÍA AL SERVICIO DEL HOMBRE"

DIRECTOR DE TESIS

DR. JOSÉ ISIDRO HERNÁNDEZ VEGA
DOCTORADO EN INGENIERÍA CON ORIENTACIÓN EN TECNOLOGIAS DE LA INFORMACIÓN
CÉDULA: 12058578

CO-DIRECTOR DE TESIS

M. C. ELDA REYES VARELA
MAESTRÍA EN CIENCIAS EN COMERCIALIZACIÓN DE LA CIENCIA Y LA TECNOLOGIA
CÉDULA: 09093449

REVISOR

ING. PEDRO RODRIGUEZ LOPEZ
INGENIERO ADMINISTRADOR DE SISTEMAS
CÉDULA: 0824645

REVISOR

ING. LUIS ALEJANDRO REYNOSO GUAJARDO
INGENIERÍA EN SISTEMAS COMPUTACIONALES
CÉDULA: 5157710

c.c.p Departamento de Sistemas y Computación
c.c.p. Expediente
c.c.p. Interesados

RESUMEN

Cuando se cuenta con una idea de proyecto para el entorno móvil es de suma importanciaconocer y plantear cada una de las fases y conceptos involucrados para realizar buenas prácticas en desarrollo de aplicaciones móviles.

Tomando como punto de partida el planteamiento del ciclo de vida que tendrá unaaplicación móvil, donde los involucrados participan en la mayoría de fases, como diseñadores de interfaces UI/UX y desarrolladores back-end y front-end.

Los desarrolladores y diseñadores realizan un análisis sobre las herramientas y entornas de desarrollo adecuadas que se utilizaran para obtener una mayor eficacia y eficiencia en el entorno de trabajo y entregar una aplicación de calidad, así mismo se definen las plataformas y publico al cual va dirigida la aplicación.

Al finalizar un proyecto es de suma importancia seguir con el ciclo de vida ya antes planificado, ya que servirá para futuros mantenimientos y actualizaciones.

Esta propuesta de investigación pretende realizar una evaluación de herramientas para la construcción de aplicaciones móviles en sus diversas etapas del desarrollo del software, aplicándolo a un caso práctico de desarrollo.

Palabras clave:

Tecnologías de desarrollo móvil hibrido, contaminantes criterio, procesamiento de variables, monitoreo, Ionic, Angular, Json, Android, Figma, Fluid, UX, UI, Google Play, Laravel, planificación, requerimientos, análisis de requerimientos, prototipado, diseño, desarrollo, testing, publicación, herramientas GUI.

ABSTRACT

When you have a project idea for the mobile environment, it is very important to know and discuss each of the phases and concepts involved to carry out good practices in mobile application development.

Taking as a starting point the approach of the life cycle that a mobile application will have, where those involved participate in most of the phases, such as UI / UX interface designersand back-end and front-end developers.

The developers and designers carry out an analysis on the appropriate development tools and environments that will be used to obtain greater effectiveness and efficiency in the workenvironment and deliver a quality application, as well as the platforms and public to which the application.

At the end of a project, it is extremely important to continue with the life cycle already plannedbefore, as it will help in future software maintenance and updates.

This research proposal aims to carry out an evaluation of tools for the development of mobile applications in its various stages of software development, applying it to a practical development case.

Key Words

Hybrid mobile development technologies, criteria pollutants, variable processing, monitoring, Ionic, Angular, Json, Android, Figma, Fluid, UX, UI, Google Play, Laravel, planning, requirements, requirements analysis, prototyping, design, development, testing, publishing, GUI tools.

AGRADECIMIENTOS

Doy agradecimientos a mi familia que nunca ha dejado de apoyarme, en alentarme, cuidarme y aconsejarme.

A mis compañeros de carrera por brindarme consejos y siempre tener su apoyo en cada semestre el cual me ha ayudado a salir adelante en momentos difíciles.

A mis tutores por apoyarme en la realización de mi trabajo de tesis, y siempre brindarme desus consejos para la mejor elaboración del documento.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE TABLAS

CAPITULO I. INTRODUCCIÓN.

1.1 Motivación del problema

Nuevo León, el estado reconocido como unos de los mejores para obtener un buen empleoen México, dado que es la sede con amplios grupos industriales y financieros; con diferentesíndoles.

Cuenta con una población de 5.1 millones de habitantes, teniendo como municipio Monterrey el más poblado con 1.1 millones de habitantes censado el 2015 (INEGI, 2015), y otros 50 municipios más involucrados.

El monitoreo de contaminantes ambientales que se realiza, indica que el Área Metropolitana de Monterrey es una de las más contaminada en México rebasando las normas de calidadde aire que establece la Secretaria de Salud, donde indica que las partículas pequeñas menores a 10 micras y partículas menores de 2.5 micras donde contienen nitratos, sulfato carbón, cuentan con un alto riesgo a la salud.

Según los registros de IQAir empresa suiza de tecnología de la calidad del aire, indican concentraciones de 86 micras-10 y de 36 micras-2.5, siendo una clara representación del daño ambiental y de la salud.

Es por esta razón que se debe de dar a conocer la información necesaria de una manera en la cual sea entendible para cualquier persona.

El uso de herramientas de Tecnologías de Información, así como la infraestructura de la Red Internet, ha hecho que se cuente con mayor disponibilidad de la información. Hoy en día la población cuenta con un teléfono inteligente que le permite enlazarse a esta red y contar con servicios digitalizados que contribuyen a realizar sus actividades de manera rápida y flexible.

Estos dispositivos móviles han generado un nicho de marcado en el desarrollo del software, adaptando el ciclo de desarrollo del software a estas nuevas necesidades de aplicaciones. Las herramientas que ayudan a construirlo no son la excepción y hoy en día contamos conuna gran variedad de herramientas de licencia de código abierto y otras de código cerrado.

La selección de herramientas no es tarea fácil, debido a que demos de conocer su objetivo, características, ventajas, desventajas, alcances y limitaciones, pero principalmente probarlas en la práctica su efectividad para contribuir a desarrollos de aplicaciones móvilesseguras y confiables una vez liberadas para el usuario final.

Seleccionar la herramienta adecuada implica analizar y comparar las existentes en el mercado, algunas implementan paradigmas de desarrollo de software de manera integral desde el análisis de requisitos hasta la etapa de pruebas de la aplicación. En sistemas de tiempo real estas aplicaciones pueden cambiar sus requerimientos, debido al tiempo de respuesta rápido, el problema del monitoreo de contaminantes ambientales implica tener información para el usuario en el momento oportuno. El presente estudio pretende hacer una aplicación móvil hibrida aplicada al monitoreo de contaminantes.

Con el análisis de herramientas para el desarrollo de aplicaciones móviles hibridas se pretende buscar herramientas actuales que existen en el mercado que permitan desarrollarla aplicación para el monitoreo de contamines, así como probar su efectividad.

1.2 Planteamiento del problema a resolver

Descripción del problema

El proyecto maneja una red de sensores los cuales monitorean los contaminantes ambientales criterio en una zona del Área Metropolitana de Monterrey la cual está próximaa una zona industria

Se cuenta con una base de datos la cual se encuentra en formato digital, lista para ser procesada y exportada la información, sin embargo, mostrar datos crudos al usuario solo causaría confusión ya que la mayoría desconoce del significado de cada contaminante y sudaño en parámetros fuera de los normales, por esta razón es necesario procesar los datospara ser explotados por la aplicación móvil.

La aplicación móvil pretende tener un diseño y lógica sencilla, obteniendo como resultado una experiencia de usuario muy eficaz y eficiente, de esta manera el usuario conocerá la información necesaria de los contaminantes que hay a su alrededor y tener la oportunidad de tomar precauciones ante altos niveles de contaminación o tomar medidas ante alguna contingencia y normas necesarias de acuerdo al caso que se tenga.

1.3 Antecedentes

Conforme han pasado los años se han desarrollado distintas aplicaciones móviles para mostrar la calidad del aire, algunas solo muestran en una determinada región y otras en más de una región.

Algunas son más completas que otras, pero cumplen con el objetivo de informar siguiendolas normas de salud.

Aire NL

Aplicación móvil la cual comunica de manera horaria a la población metropolitana de Monterrey, la calidad del aire, condiciones meteorológicas y el pronóstico de los niveles máximos esperados de material particulado y ozono durante las siguientes 24 horas.

El reporte desplegado es correspondiente al sitio del monitoreo más cercano al usuario, sinembargo, la información se puede recopilar de cualquier sitio perteneciente a la red de Sistema Integral de Monitoreo Ambienta (SIMA).

Además, es capaz de mostrar el índice IMECA (Índice Metropolitano de la Calidad del Aire)y se despliega correspondientemente al mayor.

Disponible para Android & iOS.

Aire y Salud NL

Aplicación móvil la cual permite informar acerca del Índice del Aire en el estado de Nuevo León, se presenta en diferentes calidades (Buena, Aceptable, Mala, Muy Mal y extremadamente Mala), además muestra sobre el nivel de riesgo los probables daños quepueden causar dependiendo el nivel de la calidad del aire, y las recomendaciones de las

acciones a adoptar, las cuales buscan reducir la exposición, no únicamente se busca informar a la población de la calidad del aire, sino también la opción de actuar para protegersu salud, lo cual se precisó en el objetivo de la presente Norma Oficial Mexicana (**NOM- 172-SEMARNAT-2019**), sin alterar el propósito original de dicho índice.

Disponible en Android.

Aire

Esta aplicación da información horaria sobre el Índice del Aire y Salud para mostrar losriesgos asociados a los niveles de contaminación.

Se da en las 16 alcaldías de CDMX y los siguientes municipios del Estado de México: Chalco, Valle de Chalco, Ixtapaluca, La Paz, Nezahualcóyotl, Chimalhuacán, Chicoloapan, Atenco, Acolman, Ecatepec, Tecámac, Coacalco, Tultitlán, Jaltenco, Tonanitla, Nextlalpan, Teoloyucan, Melchor Ocampo, Cuautitlán, Cuautitlán Izcalli, Tultepec, Tepotzotlán, Tlalnepantla, Atizapán de Zaragoza, Naucalpan, Huixquilican y Texcoco.

Además, es capaz de informar acerca de la intensidad de la radiación solar y ultravioleta ytiempo meteorológico.

Presentan Reportes por alcaldía, reporte por municipios, mapas de contaminantes y el programa “Hoy no Circula”.

Presentación de vectores de viento y precipitaciones en el mapa principal de la calidad deaire por estaciones.

Calidad del Aire | AirVisual

La aplicación en cuestión muestra toda la información relacionada con la calidad del aire, es posible conocer el pronóstico del tiempo y entender los riesgos que conlleva para la salud.

Actualmente se cuenta con información de más de 10,000 ciudades en más de 100 países distintos.

Brinda la oportunidad de ver datos históricos y en tiempo real los pronósticos del tiempo son a 3 días, recomendaciones para cada nivel del aire incluyendo para grupos sensibles, Mapa mundial, supervisión de los 6 contaminantes en tiempo real.

Las estaciones que se presentan son fiables y de confianza.

1.4 Hipótesis

Los frameworks para aplicaciones móviles hibridas cumplen con la misma calidad del proceso de desarrollo de software con respecto a frameworks de desarrollo móvil nativo.

1.5 Objetivos

1.5.1 General

Analizar herramientas actuales para el desarrollo de aplicaciones móviles hibridas aplicando un desarrollo de software en un sistema de tiempo real para el seguimiento de contaminantes criterio.

1.5.2 Específicos

- Investigar herramientas disponibles para el desarrollo de aplicaciones móviles hibridas.
- Realizar análisis las herramientas investigadas.
- Investigar el ciclo de vida de un proyecto de software.

1.6 Justificación del proyecto

Realizar un análisis de herramientas actuales para el desarrollo de aplicaciones móviles hibridas e implementar el desarrollo del software para llevar el seguimiento de contaminantes criterio, es parte fundamental para contar con desarrollos de software de calidad.

Para el análisis de herramientas es importante conocer su documentación técnica para versus requerimientos, ventajas, desventajas y características así planear un buen desarrollode software de aplicaciones móviles.

Una evaluación de herramientas permitirá contar con una basa de conocimiento para el desarrollo de aplicaciones móviles hibridas, contribuirá a desarrollar aplicaciones de mayorcalidad, con la capacidad de un desarrollo con información más clara y concisa para el usuario.

1.7 Impacto o Beneficio en la solución a un problema relacionado con el sector productivo o la generación del conocimiento científico o tecnológico.

Este proyecto busca vincularse con la Secretaria de Desarrollo Sustentable del Estado de Nuevo León a través del Sistema Integral de Monitoreo Ambiental (SIMA), la cual tiene la finalidad de contar con información continua y fidedigna de los niveles de contaminación ambiental en el Área Metropolitana de Monterrey.

El desarrollo de una aplicación móvil para facilitar el monitoreo de la calidad del aire, de talmanera que sea entendible para todo público, y en caso de tener cierto nivel se puedan tomar las precauciones sugeridas según las normas establecidas de salud.

Se contribuye a la formación como Ingeniero en Sistemas Computacionales dentro del áreade desarrollo de aplicaciones móviles, la cual se hará la planeación y organización del proyecto.

Se destaca el conocimiento que aportará en cuanto realizar diseño para aplicación móvil y el aprendizaje de tecnologías para desarrollo de aplicaciones móviles.

1.8 LUGARES DONDE SE DESARROLLA EL PROYECTO

El proyecto se realizará dentro de las instalaciones del Instituto Tecnológico de Nuevo Leónen su fase experimental.

1.9 INFRAESTRUCTURA

Para el proyecto se hará uso de la biblioteca digital del Instituto Tecnológico de Nuevo León(https://itnl.bibliotecasdigitales.com/?category=3) para consulta de información sobre el tema. Bases de Datos para consulta de artículos de investigación del Consorcio Nacional de Recursos de Información Científica y Tecnológica, CONRICYT. (https://www.conricyt.mx/). Se hará uso de los laboratorios de Centro de Cómputo y su infraestructura de red de computadora una vez que se regrese al instituto en semáforo verdedespués de la pandemia. Uso de Servidores de Archivos ubicados dentro del I. T de NuevoLeón. Se cuenta con la documentación oficial de las Tecnologías de desarrollo móvil: Ionic(https://ionicframework.com/docs) y Angular (https://angular.io/docs).

CAPITULO II. MARCO TEÓRICO.

2.1 El ciclo de desarrollo del software para aplicacionesmóviles.

El desarrollo de aplicaciones móviles pude parecer sencillo, sin embargo, es importante formalizar el ciclo de vida que nos permitirá dividir el proceso que conlleva conseguir la versión final de la aplicación.

Al establecer un método que guie el desarrollo de la aplicación, dará más facilidad de evaluar cada parte del proceso, evitando errores de los cuales se pueden caer si no se lesdestino el tiempo suficiente a cada fase del proyecto.

Los procesos declarados en cada fase deben cumplir con el tiempo establecido y costos estimados al inicio.

El ciclo de vida de software se compone por varias fases iniciando con planificación dondese realizan los requerimientos técnicos, prototipado, desarrollo e implementación de código,testing, publicación y mantenimiento.

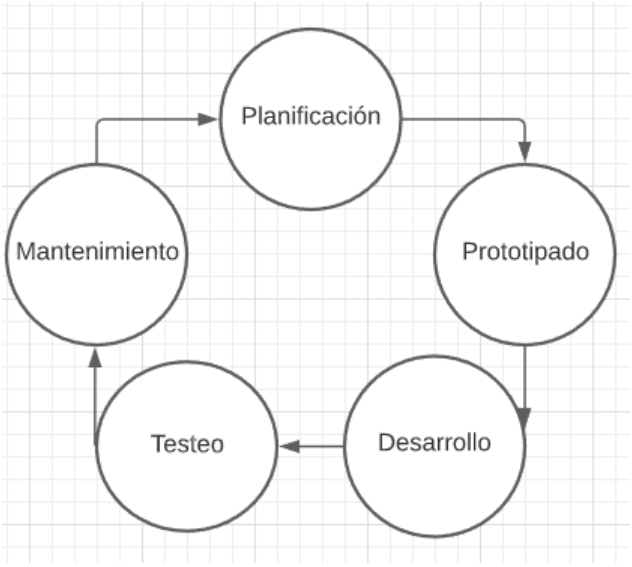

Figura 01. Ciclo de vida del proyecto.

2.1.1 Planificación

La planificación es la fase inicial del ciclo de vida del desarrollo de software móvil, se definen claramente lo que se necesitara y como se llevara a cabo el desarrollo de la aplicación.

Se evalúa la idea principal y el fin que tendrá la aplicación, ademá s se define el público al cual va dirigida.

Es de suma importancia conocer y entender el público al que estará dirigida la aplicación para enfocar completamente la estrategia posterior a lanzamiento.

A la hora de realizar un análisis de estudio de la competencia, llega la hora de tomar la decisión de definir en qué plataformas estará disponible.

2.1.2 Requerimientos técnicos

En esta fase se debe de incluir la documentación del software en la cual se detallarán todos los requisitos y su análisis dentro del proyecto.

2.1.2.1 Análisis de requerimientos

Al realizar el análisis sobre las funciones que debe cumplir el proyecto se han destacado tres puntos muy importantes.

Requerimientos Funcionales

Se refieren a la descripción de las actividades y servicios que debe de realizar y proveer elsistema. Se vincula con las entradas y salidas de procesos y datos que son almacenados al sistema, y en este caso sobre los datos que se obtendrán del servidor.

Requerimientos no Funcionales

Se refieren a las limitaciones y características que presenta el sistema, se vincula con el rendimiento, usabilidad, presupuestos, documentaciones, seguridad y tiempo de entrega.

2.1.3 Prototipado

En esta fase el responsable de crear un prototipo es un diseñador de UX/UI, el cual definepaso a paso la apariencia que tendrá la aplicación, así mismo la navegación de laaplicación.

Desde el boceto en papel a herramientas para la GUI y posteriormente la simulación de navegación.

Llevando a cabo esta fase es posible analizar todos los casos de uso, detectar inconsistencias o errores en la idea principal y corregirlo.

2.1.3.1 Experiencia de usuario (UX)

La experiencia de usuario es de suma importancia en una aplicación móvil y tener bien definida esta parte conlleva a conocer las necesidades de los usuarios a los cuales está dirigida la aplicación.

Es necesario conocer cada aspecto el cual será de utilidad para el usuario, es decir, solo colocar los elementos y acciones necesarias que satisfagan al usuario, ni más ni menos, puede que una pantalla contenga componentes muy detallados, pero hay que tener cuidadocon la información de más e innecesaria, se debe tener un control, y colocar lo que realmente y únicamente el usuario necesita.

Colocar información innecesaria podría hacer al usuario cerrar la aplicación o confundirlo.En pocas palabras la Experiencia de usuario es ponerse en los zapatos del usuario.

2.1.3.2 Interfaz de usuario (UI)

La interfaz de usuario es únicamente la apariencia la cual contiene elementos visuales, una arquitectura de información y patrones de interacción con propósitos en específico.

Es importante tocar este punto ya que, si no se logra diseñar una buena interfaz, existen las posibilidades que el usuario no logre encontrar lo que busca, si se dispone por una grancantidad de datos es importante colocar filtros entre pantallas y tener un punto fijo para losmenús, de preferencia tener un diseño dinámico y atractivo para el usuario.

La interfaz de usuario va de la mano con la experiencia del usuario.

2.1.3 Desarrollo e implementación de código.

En esta fase entran los desarrolladores, todos los requerimientos que se han detallado sobre la aplicación serán seguidos para su posterior desarrollo.

En el mundo del desarrollo implican dos partes muy importantes:

Desarrollo front-end: Los desarrolladores son encargados de realizar el código el cualmuestre la interfaz de usuario que vera el cliente, la capa con la que interactuara el usuario.Su forma es Cliente-Interfaz.

Desarrollo back-end: Los desarrolladores son encargados en realizar conexiones y peticiones con un servidor o base de datos, el cual conecta con el front-end de la aplicacióny muestra los datos.

2.1.5 Testing

Las pruebas se lanzan por tres maneras:

Tiempo real con navegador:

Mientras se programa se pueden visualizar los cambios realizados a la aplicación en tiemporeal, además de los errores que pueden ocurrir.

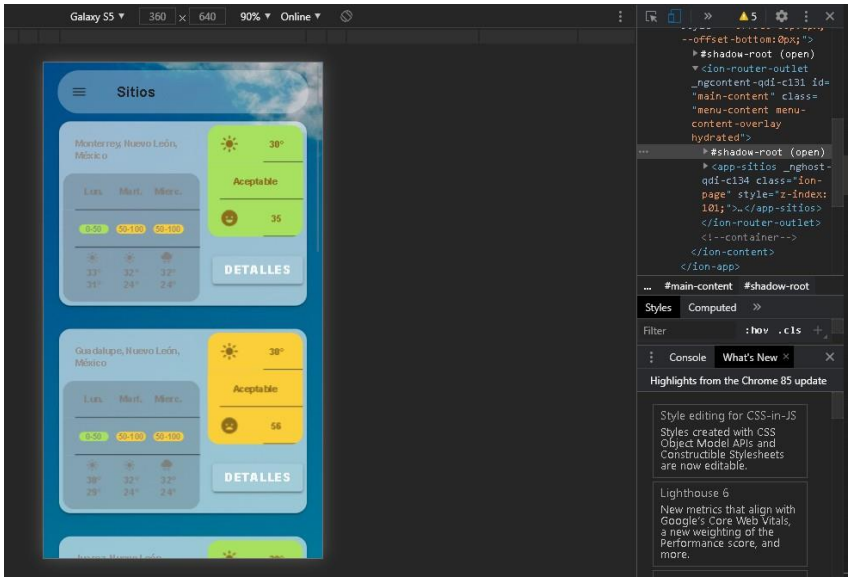

Figura 02. Testing.

Es posible depurar la aplicación de esta manera.

Teléfono físico:

Esta opción se define como la depuración USB, la cual debe estar activada esta opción enel dispositivo móvil y conexión USB a la computadora.

La finalidad es que el desarrollador pruebe el proyecto en tiempo real.Los

pasos para activar este modo son:

- En ajustes de Android entrar en “Acerca del dispositivo”.
- Buscar el “número de compilación” se debe de pulsar 10 veces seguidas, enseguida no aparece un mensaje “eres un desarrollador”.
- Regresamos a ajustes y se agregó una nueva opción “ajustes de desarrollador”
- Entramos a esa opción y activamos “Modo depuración”.

Emular un teléfono con Android Studio:

Android nos permite realizar una emulación de un teléfono con ciertas características delpropio teléfono emulado.

Existen una diversidad de teléfonos para emular con sus versiones de Android.

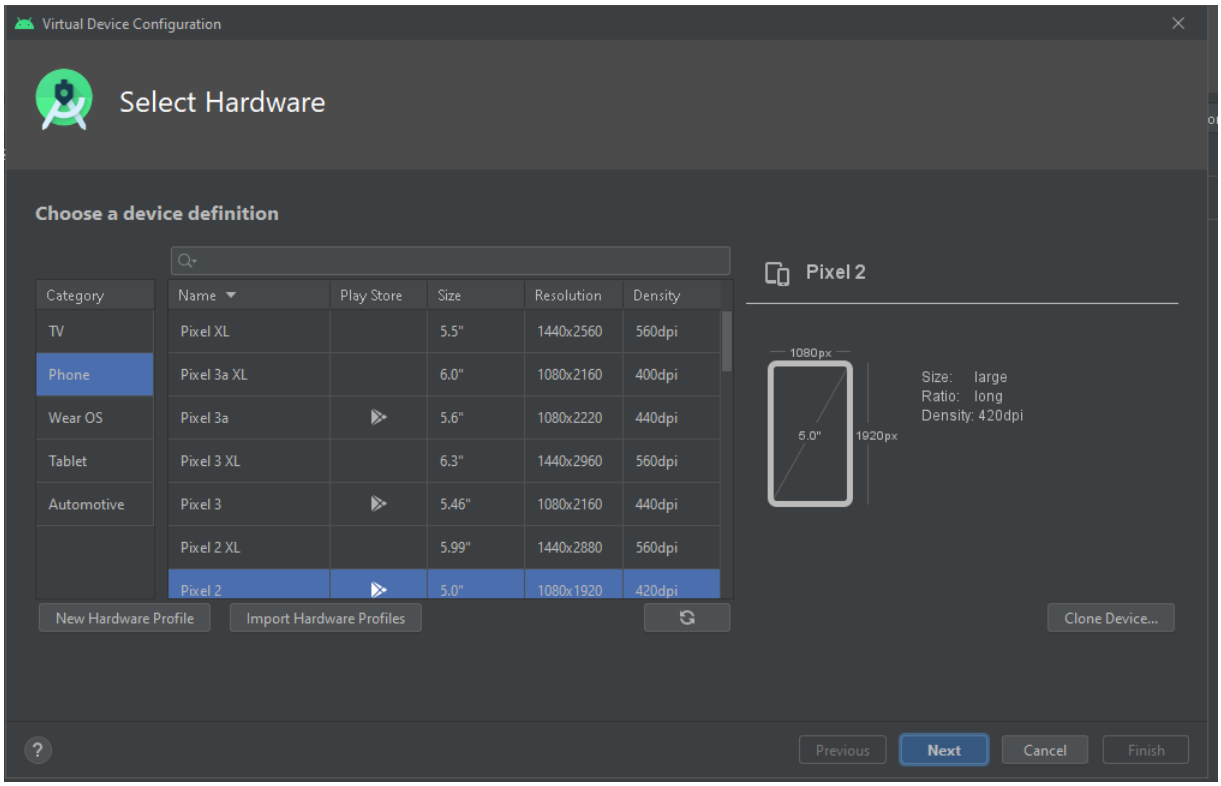

Figura 03. Dispositivos Disponibles para Emulación.

2.1.6 Publicación y mantenimiento de la aplicación.

La aplicación se construye principalmente en Android Studio de esta manera se obtendráel archivo APK.

Para realizar una publicación en Google Play es necesario cumplir con los siguientes requisitos:

- Se requiere que la aplicación este firmada digitalmente con un certificado para su posterior instalación.
- Para versiones Android 2.2 anteriores el tamaño máximo de la aplicación es de 50MB.
- Para versiones Android 2.3 y versiones posteriores el tamaño máximo de la aplicación es de 100MB.
- Debe contener un archivo manifest con la versión de la aplicación, el valor máximo permitido es de 2100000000, si se supera este número Play Store no permitirá enviar un nuevo APK.
- En el archivo manifest debe estar declarada la versión requerida con el parámetro “targetSdkVersion” y para el mínimo “minSdkVersion”, las cuales son aceptadas para ejecutar la aplicación.

Configurar la ficha de Play Store

Tabla 01. Detalles del producto

Campo	Descripción	Límite de caracteres	Notas
Titulo	Nombre de la app en Google Play.	50 caracteres	Es posible colocar un título para cada idioma.
Descripción Breve	Texto el cual se muestra a los usuarios y da un resumen de la app	80 caracteres	Los usuarios son capaces de expandir este texto para ver la descripción completa.
Descripción Completa	Describe detalladamente la app.	4,000 caracteres	

Recursos Gráficos:

En este apartado se agregan capturas de pantalla o videos que demuestren la funcionalidadde la app. Esto atrae a más público.

Idiomas y traducciones:

Al subir una aplicación el idioma por defecto es inglés, sin embargo, es posible agregartraducciones de la información de la aplicación.

Se da la posibilidad de subir videos e imágenes en diferentes idiomas utilizando los recursos gráficos.

Si no se sube alguna traducción, es posible que los usuarios vean la ficha en el idiomapredeterminado o una traducción de Google Traductor.

Categorización:

Es necesario la categoría en la cual pertenece la aplicación, si es una app o juego.

Detalles del contacto:

En este apartado se agrega información del desarrollador, como sitio web, correo electrónico, teléfono, es necesario contar con un correo electrónico de contacto para hacerla publicación.

Política de privacidad:

Si se agrega una política de privacidad se da la información acerca de cómo se administranlos datos importantes del usuario y del dispositivo.

Finalmente se configura si la aplicación será de paga o gratuita, y los países los cuales será distribuida y permitida la aplicación, se categorizan por versiones “Alpha”, “Beta” y de producción.

Mantenimiento

Cuando se lanza la versión final del proyecto y es publicada, tiene posibilidades de seguirevolucionando cuando en el ciclo de vida del proyecto.

Las mejoras son cambios en el software, puede ser visual o alguna funciona nueva.

Corrección y control de errores.

Un punto importante en el desarrollo de una aplicación, prácticamente ninguna aplicación es perfecta, pueden surgir errores, puede ser en el sistema operativo o algo en concreto de un determinado dispositivo.

Adaptación a sistemas operativos.

Cuando un sistema operativo lanza una nueva versión, generalmente cambian el cómo sehacen las cosas, por esta razón siempre se debe estar al pendiente de los cambios que serealicen, por consiguiente, se debe hacer cambios en el estilo y la adaptar la aplicación.

2.2 Herramientas orientadas al desarrollo de aplicaciones móviles.

El propósito general de las herramientas para desarrollo de aplicaciones móviles, incluyenKits de desarrollo de Software (SDK) permiten desarrollar aplicaciones para la plataforma Android, iOS y Windows.

Los internos de desarrollo integrado (IDE), permiten la agilización del flujo de trabajo.

2.2.1 Plataforma nativa SDK

Las SDK generalmente consisten en brindar fuentes núcleo de código compilado, apoyo en bibliotecas de software y componentes nativos, los cuales están diseñados para facilitar eldiseño y desarrollo de aplicaciones.

2.2.2 Android SDK Y NDK

El SDK de Android está compuesta por herramientas de desarrollo, las cuales comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales.

El NDK está conformada por un conjunto de herramientas que permiten usar e integrar código C y C++ con Android, además incluye bibliotecas que permiten acceder a las funcionalidades nativas y acceder a los componentes físicos de los dispositivos, tales comosensores y entrada táctil.

2.2.3 iOS SDK

El SDK de iOS está compuesta por un conjunto de herramientas que permite desarrollar aplicaciones móviles en el sistema operativo iOS de Apple.

Incluye un simulador de iPhone para imitar el aspecto del dispositivo en el ordenador durante el desarrollo, además permite a los desarrolladores acceder a distintas funciones yservicios de los dispositivos iOS, como atributos de hardware y software.

No se encuentra disponible para PCs con Microsoft Windows.

2.2.4 Android Studio

Proporciona las herramientas necesarias para crear aplicaciones en todo tipo de dispositivoAndroid.

Cuenta con editor visual el cual permite crear restricciones sobre su ConstraintLayout, de este modo da la posibilidad de de ajustar el tamaño de componentes y de la ventana.

Cuenta con un analizador de APK que permite encontrar posibilidades de reducir el tamañode la aplicación.

Permite trabajar con mayor productividad y agilidad, con el autocompletado y ayuda de código.

2.2.5 Xcode

El IDE para iOS es Xcode que incluye el iOS SDK y una cantidad de herramientas.

Incluye un editor de código potente, integrado en la interfaz de builder, emulador de dispositivos avanzados y documentación completa y actualizada con su versión correspondiente.

Es obligatorio utilizar la IDE Xcode si se va a publicar aplicaciones de iOS en la App Storede Apple.

Xcode es el encargado de firmar las aplicaciones y administrar las claves de desarrollador,además el proceso de verificación para su posterior presentación de la aplicación.

2.2.6 Visual Studio Code

Visual Studio Code es la IDE desarrollada por Microsoft, ayuda a los desarrolladores a encontrar el camino alrededor de una base de código extensa, con características tales como dar un vistazo a la definición y mejora GoTo.

Además, permite la codificación de aplicaciones multiplataforma en Javascript.

Consta de tres versiones dependiendo del presupuesto, Comunidad, Professional y empresarial.

Es compatible con cualquier plataforma tanto para Windows, Linux y iOS.

2.3 Introducción a tecnologías de desarrollo móvil hibrido

Al momento de realizar la planeación de una aplicación móvil, se debe de tener muy en claro que tipo de aplicación se desarrollara, al desarrollar una aplicación móvil nativa conllevaría un mayor costo de producción y la curva de aprendizaje es más alta.

Sin embargo, al desarrollar una aplicación móvil hibrida se tiene una curva de aprendizaje más sencilla, hay que mencionar que al ser hibrida se da la posibilidad de reutilizar códigoy componentes para diferentes plataformas, como Android, iOS y web. Además, es posibleutilizar recursos nativos del dispositivo como los son la cámara, desbloqueo por huella digital, GPS, entre otras.

Cuando se inició este concepto la optimización en los dispositivos era muy mala, tenía tantalentitud, sin embargo, conforme pasaron los años se ha mejorado la optimización y se ha convertido cada vez más una manera más factible para el desarrollo de aplicaciones móviles.

2.4 Frameworks de desarrollo hibrido.

2.4.1 Ionic

Ionic Framework, nos ofrece un conjunto de herramientas de código abierto, las cuales nospermitirán crear interfaces de usuario móvil, con experiencias de aplicaciones web o nativasmultiplataforma de gran calidad.

Cuenta con componentes reutilizables y de buen aspecto, los cuales son altamentecompatibles ante cualquier plataforma móvil.

Si se busca desarrollar un proyecto el cual incluya distintas plataformas es el indicado ya que será posible utilizar un mismo código para distintas plataformas.

Facilidad para integrarse a cualquier otro Framework de esta manera se ajusta para cualquier necesidad que se tenga, incluyen Angular, Vue, React o ninguno que esté relacionado con Javascript vanilla.

En caso de requerir o utilizar alguna funcionalidad nativa, Ionic da la facilidad para accedera ellas, es posible elegir la amplia biblioteca de más de 120 complementos de dispositivosnativos, dando acceso a la cámara, Bluetooth incluso el GPS.

2.4.1.1 Instalación

Inicialmente es necesario instalar Node.js, debido a que cuenta con un manejador de paquetes llamado “npm” el cual es necesario para instalar la mayoría de las dependenciasque necesitaremos.

Primero se debe descargar el instalador de Node.js desde su sitio oficial, para corroborar sise instaló correctamente y conocer que versión fue instalada se deben de ejecutar los comandos en la consola “node -v npm -v”

Enseguida de ello es de suma importancia descargar Git será necesario para tener un control de versiones de nuestros proyectos y es necesario para descargar las librerías

necesarias de Ionic para ser ejecutado, es posible descargar su instalador desde su sitio oficial.

Es necesario instalar El CLI de Ionic el cual es un complemento del Cordova CLI el cual añade una serie de características adicionales.

Finalmente se procede a incorporar Ionic Framework, para este paso es necesario ingresaren la terminal en modo administrador e ingresar el comando “npm install -g ionic/cli”, “-g” nos servirá para una instalación global y acceder desde cualquier directorio.

Para corroborar cual versión se instaló se debe ejecutar el siguiente comando en la terminal“ionic v-
“.

2.4.1.2 Comandos

La interfaz de línea de comandos de Ionic es una herramienta de referencia que nos ayudaen el desarrollo de aplicaciones Ionic.

Para poder ejecutar los comandos sobre el proyecto creado es necesario ejecutarlo dentrode la carpeta del proyecto.

Principalmente para crear un nuevo proyecto en Ionic es necesario lanzar un comando en la terminal “ionic start nombre-del-proyecto” nos creará una carpeta con el nombre del proyecto y adentro contendrá las herramientas necesarias del proyecto inicial.

Pero antes de crearse dicha carpeta nos pedirá el Framework con el que nos gustaría trabajar en conjunto a Ionic, es este caso se a elegido Angular, por mayores ventajas y comodidad.

La manera la cual se lance un proyecto es mediante el comando ejecutado en la terminal “ionic serve” el cual nos mostrare en tiempo real en el navegador las modificaciones que serealizan al proyecto.

Al añadir una plataforma es necesario ejecutar el siguiente comando en la terminal “ionic cordova plataform add Android”, de esta manera se agrega la plataforma Android.

Cuando se busca compilar una aplicación en Ionic únicamente para plataforma web se ejecuta el siguiente comando en la terminal “ionic build” de esta manera se creara una carpeta “www” la cual son activos web, enseguida de esto se construye en la plataforma móvil deseada, en este caso es Android, el comando a ejecutar en la terminal seria “ionic cordova build Android” y al ejecutarlo se creara la carpeta app la cual contendrá los archivosnecesarios para poder construir una apk en Android studio.

La manera de probar el proyecto en dispositivo es necesario ejecutar el comando “ionic runandroid” en la terminal y enseguida se ejecutará la aplicación en un dispositivo conectado.

2.4.1.3 Estructura del proyecto

La estructura de un proyecto de Ionic se puede definir gráficamente de la siguiente manera.

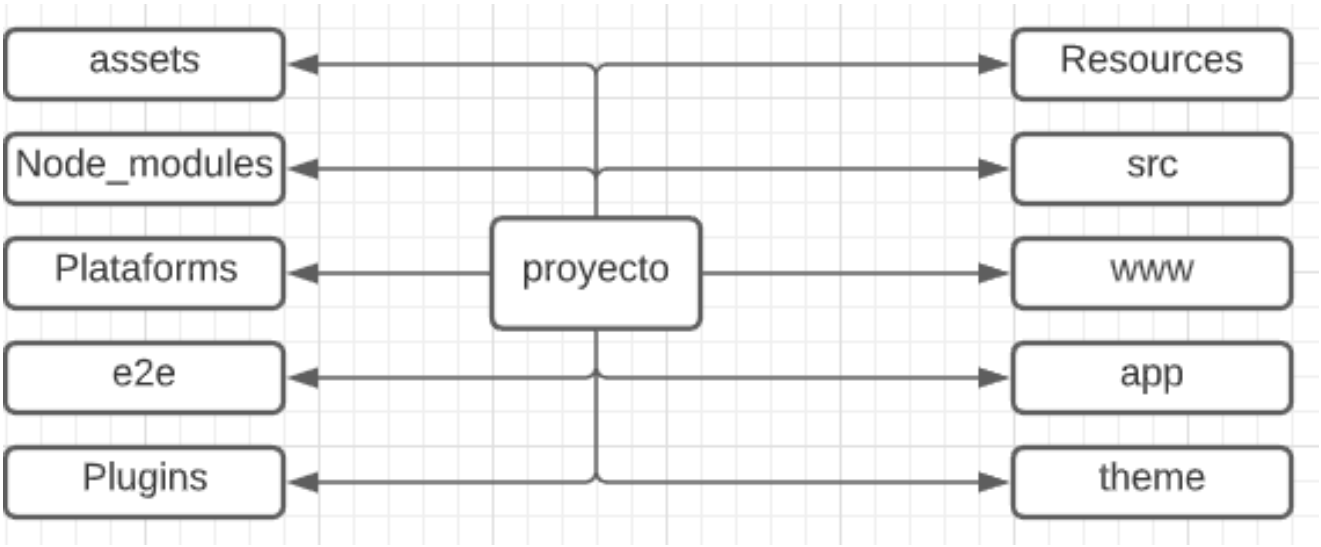

Figura 04. Estructura general de un proeycto Ionic.

e2e: Esta carpeta tiene el significado “end to end” sirve para añadir y gestionar las pruebasde nuestro proyecto.

En si es una metodología donde se evalúa de principio y a fin, se realizan simulaciones encasos de uso real y realizando integraciones de los componentes y sistema.

Node_modules: Esta carpeta se compone de todos los módulos de Ionic y dependenciasque se instalen y necesiten en el proyecto.

Platforms: Esta carpeta no es incluida inicialmente, pero se crea automáticamente cuandose decida añadir alguna plataforma.

El contenido de esta carpeta es el código específico de las plataformas añadidas que se van a compila.

Plugins: El contenido de esta carpeta son plugins de Cordova instalados para sus respectivas plataformas y añadir funciones nativas del móvil. Esta carpeta se incluye al añadir una plataforma, es decir no se incluye al iniciar un proyecto.

Resources: Generalmente se utiliza para guardar los iconos e imágenes que se usaran enlas splashscreen de las plataformas específicas.

src: El código fuente del proyecto se encuentra justamente en esta carpeta.

www: Esta carpeta contiene el código del proyecto compilado. Primero se transpila el código escrito Typescript y finalmente se genera el código en Javascript que se utilizara.

App: Carpeta contenedora de los componentes principales del proyecto.

Assets: Carpeta contenedora de todos los recursos que se utilizaran en el proyecto, incluye imágenes, videos, audio, json, etc.

Theme: Carpeta contendora de las configuraciones de los temas de la aplicación.En este caso se utilizan ficheros scss generados junto a la aplicación.

2.4.1.4 Características de Ionic

Ionic cuenta con una gran cantidad de características las cuales brindan muchas ventajas.

Potente CLI

Ejecutando un solo comando será posible crear, construir, probar y compilar un proyecto en cualquier plataforma que se añadida.

AngularJS & Ionic

Ionic no solo da buen aspecto a los proyectos, gracias a integrar Angular se crea un marcomás adecuado para el desarrollo de aplicaciones con una arquitectura central robusta y seria.

Centro nativo

Lo interesante que tiene esta librería es que nos ofrece desarrollar el proyecto una vez y la posibilidad de compilar para varias plataformas, esto gracias a la inspiración que tomo Ionicen varias SDK de desarrollo nativo, da la facilidad de familiarizarse con cualquier persona que ha construido una aplicación móvil nativa.

Optimización

Ionic cuenta con un alto rendimiento, debido a su mínima manipulación del DOM en conjunto trabaja con aceleraciones de transición y no trabaja con JQuery.

Diseño agradable

Lo que Ionic ofrece los componentes más usados en móviles, tipografías, elementosinteractivos, los cuales son fáciles de manejar y darle una buena apariencia a tu proyecto.

2.5 Herramientas para la GUI para el desarrollo móvil hibrido

2.5.1 Figma

Figma es una aplicación la cual es posible diseñar interfaces de aplicaciones ya sea para web, aplicación móvil o de escritorio, además es posible realizar el prototipado de proyectos colaborativamente en tiempo real, creando grupos de trabajo para colaborar con el mismo diseño.

Está basada en navegador y muy apta para la experiencia de usuario e interfaz de usuario,además es posible generar código css.

Sistema de comentarios integrados para obtener retroalimentación y tomar decisiones correctas a futuro, dentro del prototipo es posible mencionar a miembros del equipo para notificarlos sobre los cambios o la retroalimentación obtenida.

2.5.2 Fluid UI

Fluid es una herramienta intuitiva para construir prototipos rápidos y elaborar diseños.

Se estiman más de dos mil componentes integrados, eligiendo bibliotecas como MaterialDesign, iOS, Windows y Wireframe.

Es posible crear prototipos interactivos, con toques, deslizamientos, clics, entre otrosgestos, gracias al sistema de vinculación de Fluid UI.

La función de animar el diseño creado, hace que el prototipo sea único. Ofreciendo Pop in,fade in, flip o simples show para dar animaciones que ayudan a darle vida al prototipo.

Brinda la oportunidad de crear enlaces de navegación en el propio prototipo, que ayuda amostrar cómo se relacionan las paginas entre sí.

La interfaz de Fluid ofrece crear prototipos en línea, junto a con otros miembros de trabajo,todo el trabajo realizado será guardado en la nube lo que brinda total accesibilidad para cualquier miembro del equipo.

2.5.3 Componentes de UI

Ionic cuenta con una serie de bloques de construcción de alto nivel denominados Componentes, los cuales permiten construir un diseño de la interfaz de usuario agradable y rápidamente.

ion-content: Permite controlar el área desplazable, solo es posible tener un “content” en una sola vista.

```
<ion-content ></ion-content >
```

Figura 05. ion-content.

ion-grid: Componente el cual permite crear una caja flexible y crear diseños personalizadosen dispositivos móviles.

```
<ion-grid></ion-grid>
```

Figura 06. Ion-grid.

ion-row y ion-col: Despues de colocar un grid se coloca una “row” y finalmente un “col” este es el encargado de rellenar el “row” conforme cambie el tamaño del dispositivo se ajustará la “col” en “row”, está basado en un diseño de 12 columnas.

```
<ion-row>
<ion-col>
</ion-col>
</ion-row>
```

Figura 07. Ion-row y ion-col.

ion-card: Las tarjetas son un estándar para dar detalles al usuario, se puede componer deun solo componente o puede contener, encabezados, titulo, subtitulo, imágenes y contenido.

<ion-card></ion-card>

Figura 08. Ion-card.

ion-button: Este componente es utilizado para realizar acciones al hacer clic sobre él utilizado generalmente en envió de formularios.

Es posible pasar parámetros a la función definida.

<ion-button></ion-button>

Figura 09. Ion-button.

ion-menu: Componente desplazable de forma predeterminada del lateral izquierdo, sin embargo, es posible anular esta manera. Contiene ítems los cuales en general son utilizados para navegar en la aplicación. Pueden existir más de un menú adjuntos al contenido.

<ion-menu></ion-menu>

Figura 10. ion-menu.

Modal: El modal aparece encima del contenido de la aplicación en forma de dialogo, ocupando toda la pantalla de la aplicación, es de mucha utilidad en distintos casos, sin embargo, en mi aplicación es utilizado para brindar detalles generales de la ubicación seleccionada.

Se conforma por una variedad de componentes y se define como una vista más.

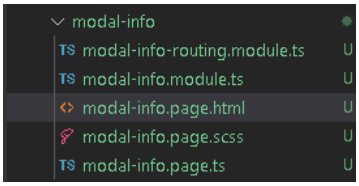

Figura 11. Modal.

ion-icon: Son utilizados para mostrar iconos específicamente diseñados para distintas plataformas.

<ion-icon></ion-icon>

Figura 12. Ion-icon

ion-item: Generalmente es usado en lista juntos otros elementos, es posible deslizar, eliminar, ordenar y editar los ítems, están compuestos por contenido como texto, imágenes,avatares e incluso elementos nativos y personalizados.

```
<ion-item></ion-item>
```

Figura 13. ion-item.

ion-toolbar: Utilizadas principalmente para colocar componentes útiles para el usuario, como botón de menú, título de la vista actual, es posible colocarlo en la cabecera o en el pie de la vista actual.

```
<ion-toolbar></ion-toolbar>
```

Figura 14. ion-toolbar.

2.5.4 Estructura de archivos de un componente o page.

Al momento de crear un componente se crean dinámicamente los archivos que conforman el componente y son necesarios para su funcionamiento.

Example.component/page.html: Este archivo es la plantilla esqueleto del componente donde se agrega componentes de UI de Ionic, y personalizados propios, se conforma de etiquetas de html.

Example.component/page.scss: Archivo de estilo de hoja, nos permite darle estilo y apariencia personalizada a nuestros componentes propios o componentes UI de Ionic.

Example.component/page.ts: Archivo donde se puede escribir lógica, sin embargo, parabuenas prácticas es recomendable importar services o pipes y llamar sus funciones.

Example.module.ts: Los módulos son los encargados de importar y exportar las librerías necesarias para obtener sus funciones, nos permite extender nuestra aplicación y tener mejor control en la organización.

Example.component/page.spec.ts: En este archivo nos brinda la facilidad de realizar pruebas unitarias de funciones que utiliza nuestro componente actual.

Example-routing.module.ts: Modulo especialmente creado para exportar las rutas de navegación a nuestras vistas, además importa las vistas para obtener su ruta y acceder a ella específicamente.

2.5.5 Sintaxis de los archivos de un componente y/o page.

Example.component.html. Plantilla HTML

```
<div clas
<h2>{{Pollutant
<p >
{{Valor}}
</p>
</div>
```

Figura 15. Sintaxis component.html.

Este compuesto de etiquetas HTML, este componente puede ser el hijo o padre de algúncomponente en específico, esto depende del objetivo del componente.

Example.page.html. Plantilla HTML

```
<app-menu></a
<ion-content >
<ion-router-outlet id="main-content"></ion-rout
</ion-content>
```

Figura 16. Sintaxis page.html.

Una página tiene una diferente sintaxis en su archivo HTML, esta debe contener ion-content, componente UI de Ionic especializado para hacer el contenido interactivo.

Además, aquí colocamos nuestros componentes creados.

Example.component/page.scss. Hoja de Estilos.

```
ion-content{
--background:url("../../assets/imgs/F_Hojas.png");
}
#elements{
display: inline-block;
}
.Element{ background: black;}
```

Figura 17. Sintaxis component/page.scss

Un componente o página utiliza el mismo tipo de archivo para darle apariencia a nuestro esqueleto html.

La lectura del archivo es descendente, e importa el orden.

ion-content: es una etiqueta componente en general, todas las etiquetas que se encuentren presentes en chivo HTML tendrán las mismas propiedades a menos que se asigne una clase.in embargo es importante declarar esa clase después de esta declaración.

#elements: al utilizar el "#" se indica que se dará estilo a una etiqueta con determinada id solo a esa etiqueta se aplicará el estilo.

.Element: al utilizar "." Al inicio indica que se declara una nueva clase, y es posible asignaresta clase a diferentes etiquetas, por ende afectara a todas ellas el estilo.

Example.component/page.ts Clase

```
import { Component, Input, OnInit } from '@angular/core';
```

Figura 18. Sintaxis component/page.ts ""import".

import: Realiza importaciones necesarias al principio del archivo, agregando la ruta de obtención de la importación.

```
@Component({
  selector: 'app-modal-info',
  templateUrl: './modal-info.page.html',
  styleUrls: ['./modal-info.page.scss'],
})
```

Figura 19. Sintaxis component/page.ts "Decorador".

@Component: Es un decorador que define al archivo como un componente.

selector: Asigna el nombre de identificación de nuestro componente, se coloca de manerade etiqueta <app-modal-info></app-modal-info>.

Es posible cambiarle el nombre, es recomendable asignar un nombre que sea simbólico y describa el componente.

templateUrl: Obtiene la Url de la plantilla HTML que se usara como esqueleto delcomponente.

styleUrls: Obtiene la Url de la hoja de estilos que se aplicara en la plantilla HTML.

```
export class ModalInfoPage implements OnInit {}
```

Figura 20. Sintaxis component/page.ts "export class".

El componente o page se exportan como clase para ser utilizado en otros componentes uobtener la vista para su ruta.

Example.module.ts Módulo

```
import { NgModule } from '@angular/core';
```

Figura 21. Sintaxis module.ts "import".

Es importante realizar la importación de "ngModule" para realizar buenas prácticas esrecomendable insertarla al inicio del archivo.

```
@NgMod
entryCompone
imports: [
CommonModule,
FormsModule,
IonicModule],
declarations: [SitiosPage],
providers:[HTTP,UsaqiPipe]
})
```

Figura 22. Sintaxis module.ts "Decorador".

@NgModule: Declaración de un decorador que indica angular que se trata de un módulo

entryComponents: Declaración de componentes de entrada, como lo son los Modales. imports:

Importa modulos externos junto a sus funciones, componentes o rutas.

providers: Declarar componentes en esta propiedad indica que serán utilizados como servicios en nuestro archivo de lógica.

declarations: Declara componentes que serán utilizados en la plantilla HTML.

```
export class SitiosPageModule {}
```

Figura 23. Sintaxis module.ts "export class".

El módulo se exporta para poder ser usado en otros módulos.

Example.component/page.spec.ts. Pruebas unitarias

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';import
{ IonicModule } from '@ionic/angular';
import { SitioComponent } from './sitio.component';
```

Figura 24. Sintaxis spec.ts "import".

Las importaciones necesarias para realizar pruebas sobre nuestro componente están plasmadas en la Figura 12. El tercer import es la importación de nuestro componente o servicio al cual se harán las pruebas.

```
describe('SitioComponent', () => {
//Lógica a testear};
```

Figura 25. Sintaxis spec.ts "Funcion describe".

Con la función especial “describe” como primer parámetro recibe nuestro componente el cual se busca probar una función propia, como segundo parámetro se ejecuta una funciónanónima, la cual ejecutara nuestra función del componente seleccionado.

Example-routing.module.ts Módulo de rutas

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
```

Figura 26. Sintaxis rotuing-module.ts “import”.

Las importaciones necesarias para un módulo de rutas son “NgModule”, “Routes” y “RouterModule”.

```
import { SitiosPage } from './sitios.page';
```

Figura 27. Sintaxis routing-modules.ts “import vistas”.

Un módulo de rutas necesita importar las rutas o ruta siendo el caso de la propia vista,asigna las rutas disponibles en las que se pude navegar.

```
const rou
{
path: '',
component: SitiosPage
}
];
```

Figura 28. Sintaxis routing-module.ts “Routes”.

Se declara una variable “routes” de tipo “Routes”, dentro del arreglo se encuentra un objetode dos propiedades.

path: asigna la ruta de la vista.

component: componente seleccionado a determinada ruta.

```
@NgModule({
imports: [RouterModule.forChilexports:
[RouterModule],
})
```

Figura 29. Sintaxis routing-module.ts “Decorador”.

@NgModule: Declaración de un decorador que indica a angular que se trata de un módulo,solo utiliza dos propiedades.

imports: Importa la ruta del componente o componentes, siendo el caso de una ruta hijo es declarado como “forChild”.

exports: Exporta el módulo de rutas, puede contener una ruta misma de la vista, o un conjunto de rutas declaradas en “routes”.

```
export class SitiosPageRoutingModule {}
```

Figura 30. Sintaxis routing-module.ts “export class”.

El módulo se exporta para ser usado en otros módulos de ruta.

2.6 Herramientas para el Modelo de Datos para el desarrollo móvil hibrido.

2.6.1 Angular.

Angular es un Framework web, web móvil, móvil nativo y escritorio el cual será posible reutilizar código en la creación de una aplicación, para cualquier tipo de objetivo de implementación.

Ofrece un amplio control en la escalabilidad y satisfacer los requerimientos grandes de data, los cuales ofrece modelos de data en RxJS, Immutable.js u otro modelo de empuje.

La declaración de funciones resulta ser rápida, añadiendo plantillas declarativas simples, es decir, ya sea un componente o pantalla es posible colocarlo y reutilizarlo en varios lugares, sin necesidad de escribir código completo de este mismo.

La obtención de ayuda y comentarios inmediatos propios de Angular son compatibles con la mayoría de los IDE y editores.

Al iniciar una aplicación únicamente cargara los componentes, vistas y funciones necesarias, obteniendo una velocidad de carga rápida.

Además, proporciona vistas para Node.js, .NET, PHP entre otros servidores, para la programación back end, realizando todo tipo de peticiones a los servidores.

2.6.2 Laravel.

Laravel es un Framework de PHP que permite el desarrollo de proyectos totalmente personalizados.

Es uno de los más utilizados y con una comunidad muy grande.

Ofrece funcionalidades potentes a los desarrolladores permitiendo agilizar el desarrollo de las aplicaciones.

Para Laravel es importante hacer énfasis en la calidad del código, la facilidad del mantenimiento y escalabilidad, permitiendo desarrollar desde pequeños proyectos a grandes, facilita el trabajo en equipo y promueve las buenas prácticas.

Laravel sigue una arquitectura basada en carpetas avanzadas, lo que promueve la separación de los archivos con un orden correcto y definido.

Además, es posible seguir la arquitectura por clases la cual promueve la separación de código por responsabilidades.

El estilo que maneja Laravel es MVC (Modelo, Vista y Controlador).

Sigue un sistema de rutas, las cuales son fáciles para crear todo tipo de URLs dinámicas para los usuarios.

2.6.3 Node.js

Ideado como un entorno de ejecución de JavaScritp orientado a eventos asíncronos, Node.js está diseñado para aplicaciones network escalables.

Incluye un bucle de eventos como runtime de ejecución en lugar de una biblioteca.

HTTP es un elemento destacado en Node.js, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Es lo que lo hace muy adecuado para la basede una librería o framework.

Node.js está diseñado para trabajar sin hilos, no significa que no pueda aprovechar múltiples núcleos en su entorno.

Es posible crear subprocesos o procesos hijos utilizando la API “child_process.fork()”, la cual está diseñada para la comunicación entre ellos sea fácil en su proceso principal.

2.6.4 Obtención de datos

Para obtener los archivos necesarios es de suma importancia realizar conexiones de protocolos HTTP para realizar la comunicación con el servidor en cuestión de descargar ocargar datos.

Angular cuenta con una API HTTP totalmente simplificada denominada “HttpClient”.Lo cual

ofrece lo siguiente:

- Manejo de errores.
- Funciones para el testing.
- Interpretar solicitudes y respuestas.
- Realizar solicitudes de objetos respuesta escritos.
-

Para poder utilizar HttpClient es necesario importarlo en el módulo donde se utilizará.

app / app.module.ts (extracto)

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  imports: [
    BrowserModule,
    // import HttpClientModule after BrowserModule.
    HttpClientModule,
  ],
  declarations: [
    AppComponent,
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

Figura 31. HTTPClient import.

Ahora es posible inyectar como una dependencia de una clase de la aplicación.

app / config / config.service.ts (extracto)

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable()
export class ConfigService {
  constructor(private http: HttpClient) { }
}
```

Figura 32. HTTPClient Service Injectable.

El HttpClient utiliza observables para todas las transacciones que se realicen.

Es por ello que se deben de importar los símbolos de operador y observables RxJs.

app / config / config.service.ts (importaciones RxJS)

```
import { Observable, throwError } from 'rxjs';
import { catchError, retry } from 'rxjs/operators';
```

Figura 33. Rxjs

Para solicitar datos se utilice el método HttpClient.get() para obtener datos de un servidor, es un método asíncrono que envía una solicitud HTTP y devuelve un observable dando losdatos que se solicitaron.

CAPITULO III. METODOLOGÍA DE SOLUCIÓN.

3.1. Metodología utilizada para la construcción de la aplicación móvil.

Se elaboró un plan para el ciclo de vida del proyecto, el cual se siguió acordea él, se especifica con detalle las actividades realizadas en cada etapa de vida del proyecto, siguiendo el siguiente esquema.

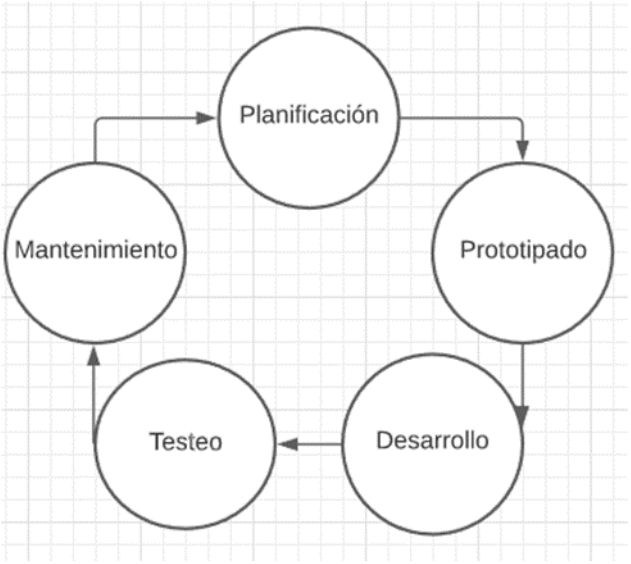

Figura 01. Ciclo de vida del proyecto.

Planeación.

Entorno de desarrollo Integrado. Visual

Studio Code.

Visual Studio Code permite escribir código y desarrollar programas, especialmente utilizado para proyectos de aplicaciones web o móviles, siendoel caso de utilizar Frameworks.

En el desarrollo del proyecto nos permite agilizar la navegación entre carpetas, además de adoptar una buena organización de nuestros componentes.

El proyecto consta de archivos de diferentes tipos de extensión, VSC nos permite abrir y editar estos archivos, para compilar y realizar pruebas en tiempo real es necesario, instalar Node.js, a través de su servidor local, el cual fue previamente instalado.

Frameworks utilizados.Ionic

Tabla 02. Evaluación de Ionic

Titulo	Descripción
Sostenibilidad	Permite la modernización y el mantenimiento en las aplicaciones realizadas.
Comunidad	Existe una amplia comunidad lo cual es importante para la resolución de posibles problemas que se puedan presentar.
Documentación	La documentación disponible en su sitio web oficial es sencilla, entendible siendo nuevo en el tema.
Curva de aprendizaje	Es más sencillo que Angular por el hecho de ser solo un Framework
Lenguaje de programación	Javascript
Desarrollo con etiqueta y estilos	HTML & CSS
Compatibilidad SO	Multiplataforma
Compatibilidad con otros Frameworks	Angular, CodeIgnite, Bootstrap, MaterialUI

El Framework Ionic es utilizado en el proyecto principalmente por ser una herramienta para interfaz de usuario, con sus componentes ya programados porel equipo de Ionic.

Pages.

Ionic está conformado de componentes UI ya creados por los desarrolladoresde Ionic, facilita y agiliza el desarrollo de la aplicación.

Componentes.

Al poseer conocimiento previo de estas tecnologías es posible modificar los componentes que ofrece, con la intención de ajustarlo al diseño de la aplicacióny tener la oportunidad de reutilizarlos.

Proporciona estabilidad, alta calidad y rendimiento al utilizar tecnología web como HTML, CSS y JavaScript.

Page.

Es encargo de registrar y visualizar paginas especificas en base a URL, esusada debajo por NavController, es decir, no interactúa directamente con él.

Al insertar una nueva página el NavController con la URL se encarga deactualizar y hacer que coincidan con la ruta de la página.

Angular

Tabla 03. Evaluación de Angular

Titulo	Descripción
Sostenibilidad	Permite la modernización y el mantenimiento en las aplicaciones realizadas, aun así, realizando unanueva versión del Framework es posible actualizarlo sin problemas de compatibilidad.
Comunidad	Existe una amplia comunidad lo cual es importante para la resolución de posibles problemas que se puedan presentar.
Documentación	La documentación disponible en su sitio web oficial es muy técnica y difícil para entender siendo el caso de ser nuevo en el tema de Framework, sin embargo, otorga una buena herramienta para las personas más avanzadas en el tema.
Curva de aprendizaje	Al ser un Freamework su curva de aprendizaje es compleja y lenta.
Lenguaje de programación	Typescript/Javascript
Compatibilidad SO	Multiplataforma
Compatibilidad con otros Frameworks	Ionic, CodeIgnite

Componentes

Angular se basa en componentes los cuales contienen algunos archivos de código por separado, el componente de TypeScript, archivo HTML y hoja de estilos, con la posibilidad de reutilizarlos.

Además, es posible crear Subcomponentes los cuales son utilizados para especificas tareas de un componente, dentro del proyecto es muy utilizado.

Módulos

Encargado de organizar las partes de nuestro proyecto, se encuentra ordenado en bloques, nos facilita la extensión de nuestro proyecto con funciones de librerías externas.

Permite saber las importaciones y exportaciones necesarias para el funcionamiento de un especifico componente.

LocalStorage

Se almacenan localmente las preferencias de usuario tales son los lugares de preferencia a mostrar en la vista “Sitios”.

HTTP

Angular nos ofrece un módulo HTTPModule y HTTPClient para realizar peticiones HTTP la cual nos facilita mucho el trabajo al hacer realizar petición a la API realizada de contaminantes y variables meteorológicas.

Estas peticiones se realizan en base de accionar un botón o programadas p ara obtener los datos en cierta hora automáticamente.

Service

Servicios implementados para mantener separado de los componentes, se utilizan principalmente para alojar código el cual hará uso de peticiones HTTP y/o lógica.

Interface

Interfaz de serie de datos para alojar en un arreglo obtenidos de una petición HTTP (**Figura 19**).

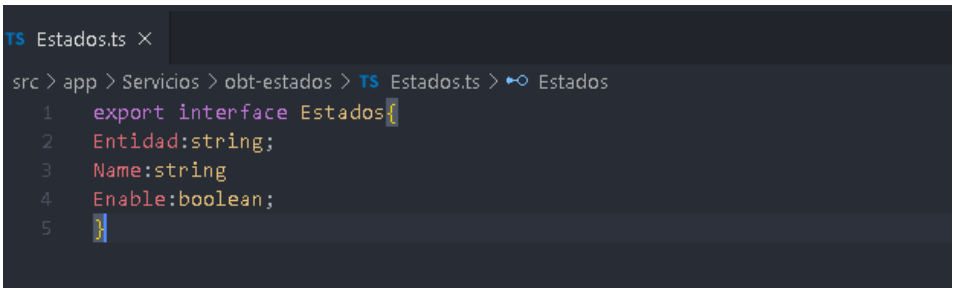

Figura 34. Interface

Directivas ng

Estructuras hechas para elementos HTML los cuales permiten añadir, manipular o remover elementos del DOM.

ngFor según recorre un arreglo obtenido de una petición HTTP realizada por el usuario o actualización automática programada al servidor para obtener el archivo JSON y los datos según la consulta con los parámetros correspondientes ya programados.

Por consiguiente, se generan componentes dinámicos en base a un componente prediseñado.

ngIF es la condición que se requiere para colocar la plantilla correcta según el índice obtenido de la petición realizada del sitio en concreto, para colocar la carita, el color y condición de la calidad del aire.

Firebase.

Servicio donde se almacenará la información en un archivo JSON con la información correspondiente de con las variables meteorológicas y contaminantes criterio.

Se realizarán peticiones a la API de FireBase siguiendo el siguiente código.

```
this.itemref = this.db.object(this.Date);let d=
this.itemref.snapshotChanges();
```

Figura 35. Código obtención de datos.

Se utiliza la función “object()”, como parámetro es el objecto que se desea obtener en estecaso se pasa la fecha actual en formato “Año/mes/dia”, siguiendo el siguiente ejemplo: 2020123, en contexto “dos mil veinte, mes doce, día tres.

```
Date = this.Year+""+this.Month+""+this.Day;
```

Figura 36. Obtencion de la fecha.

Se utiliza la variable “Date” para obtener las fechas.

Formato base de datos en Firebase.Airnl-58f75.json.

Archivo con formato JSON contiene las variables Meteorológicas y contaminantes criterio, además en la hora que fue tomada, se colocó palabra clave la ubicación para tener referencia sobre el lugar, y como clave principal y obtener todos los datos de cierta fecha es necesario ingresar por año, mes y día, el día cuando es menor a diez debe ser solo un digito sin agregar 0, ejemplo 01.


```
{
  "2020121" : [ {
    "CO" : 2.11,
    "Hora" : "00:00",
    "Humedad" : 96,
    "Location" : "Guadalupe, Nuevo León, México",
    "NO2" : 0.133,
    "O3" : 0.019,
    "PM10" : 57,
    "PM25" : 50,
    "SO2" : 0.005,
    "Temp" : 11.2,
    "Viento" : 8.1
  }, {
```


Figura 37. Estructura de archivo json.

Análisis del Problema.

No se cuenta con una aplicación móvil la cual muestre información concisa y clara sobre la condición del aire y variables meteorológicas en Nuevo León.

Lo cual conlleva a que la población de Nuevo León desconozca estos datos, de manerarápida y a la mano.

Existen algunas aplicaciones por parte de SIMA Nuevo León sin embargo no consta de unainterfaz de usuario agradable.

La segunda aplicación no muestra datos en ningún momento, solo muestra “NA”

Desarrollo de la solución. Análisis

de requerimientos. Requerimientos

de usuarios.

- **Interfaz de usuario**

Debe ser muy intuitiva y clara para el usuario sin necesidad de tener algúnconocimiento previo o leer algún manual para utilizar la aplicación.
El menú está diseñado con dos rutas una para acceder a sitios que es la vista principal, muestra la información resumida de cada ciudad de preferencia de usuarioy la otra es dudas vista que contiene el significado, los rangos y daños hacia la saludde cada contaminante.

Vista Principal/Sitios
Componente “Resumen de ciudad” que describe la calidad del aire junto variablesmeteorológicas diseñado para ser comprendido para todo público.
Contiene un botón “Ver Detalles” el cual abre un modal que muestra la informaciónen general del lugar, los índices de los contaminantes en formato de gráfica.

Vista Dudas
Los usuarios que no se encuentran tan relacionados con el significado de los contaminantes, la calidad del aire y variables meteorológicas y tengan alguna duda,será posible resolver las dudas más importantes como “Daño que causa cada contaminante hacia la salud”, “Rangos de calidad del aire y nivel de los contaminantes”.

Requerimientos del sistema.

- Es necesario contar con un equipo Android con versión 5.0 +.
- La capacidad de Procesador deberá ser 1.0 Ghz como mínimo.
- Memoria RAM debe ser igual o mayor a 2GB.
- Almacenamiento disponible debe ser mayor a 80MB.

Requerimientos legales.

Toda información acerca de los niveles de los contaminantes e índice de calidad del aire seha tomado de la NORMA Oficial Mexicana NOM-172-SEMARNAT-2019, siendo así información verídica para los usuarios.

Tabla 04. Gama de colores según la calidad de los 6 contaminantes criterio.

Índice AIRE y SALUD	Nivel de riesgo asociado	Intervalo de PM_{10} promedio móvil ponderado de 12 horas* (µg/m³)	Intervalo de $PM_{2.5}$ promedio móvil ponderado de 12 horas* (µg/m³)	Intervalo de O_3 promedio de una hora (ppm)	Intervalo de O_3 promedio móvil de ocho horas (ppm)	Intervalo de NO_2 promedio de una hora (ppm)	Intervalo de SO_2 promedio móvil de 24 horas (ppm)	Intervalo de CO promedio móvil de ocho horas (ppm)
Buena	Bajo	≤ 50	≤ 25	≤ 0.051	≤ 0.051	≤ 0.107	≤ 0.008	≤ 8.75
Aceptable	Moderado	>50 y ≤ 75	>25 y ≤ 45	>0.051 y ≤ 0.095	>0.051 y ≤ 0.070	>0.107 y ≤ 0.210	>0.008 y ≤ 0.110	>8.75 y ≤ 11.00
Mala	Alto	>75 y ≤155	>45 y ≤79	>0.095 y ≤0.135	>0.070 y ≤0.092	>0.210 y ≤0.230	>0.110 y ≤0.165	>11.00 y ≤13.30
Muy Mala	Muy Alto	>155 y ≤235	>79 y ≤147	>0.135 y ≤0.175	>0.092 y ≤0.114	>0.230 y ≤0.250	>0.165 y ≤0.220	>13.30 y ≤15.50
Extremadamente Mala	Extremadamente Alto	> 236	> 147	> 0.175	> 0.114	> 0.250	> 0.220	> 15.50

Los colores son tomados de la siguiente tabla:

Tabla 05. Tabla de códigos utilizados para la calidad de los contaminantes.

Color	R	G	B	C	M	Y	K
Verde	0	228	0	40	0	100	0
Amarillo	255	255	0	0	0	100	0
Naranja	255	126	0	0	51	100	0
Rojo	255	0	0	0	100	100	0
Morado	143	63	151	51	89	0	0

Cálculo del índice de calidad del aire

Se sigue el documento EPA 454/B-18-007 para realizar los cálculos y rangos de AQI (Calidad del aire).

Para los contaminantes criterio, se toma su concentración para realizar estos cálculos.

Tabla 06. Nombres y colores de las 6 categorías de Índice de calidad del aire.

For this AQI...	use this descriptor...	and this color
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon

Tabla de colores en formato RGB que corresponden a los rangos de calidad del aire decontaminantes.

Tabla 07. Formula de colores Índice de calidad del aire.

Color	R	G	B	C	M	Y	K
Green	0	228	0	40	0	100	0
Yellow	255	255	0	0	0	100	0
Orange	255	126	0	0	52	100	0
Red	255	0	0	0	100	100	0
Purple	143	63	151	51	89	0	0
Maroon	126	0	35	30	100	100	30

Fórmula utilizada para calcular el índice de calidad del aire en base a la concentración deun determinado contaminante.

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}(C_p - BP_{Lo}) + I_{Lo}.$$

Where I_p = the index for pollutant p

C_p = the truncated concentration of pollutant p

BP_{Hi} = the concentration breakpoint that is greater than or equal to C_p

BP_{Lo} = the concentration breakpoint that is less than or equal to C_p

I_{Hi} = the AQI value corresponding to BP_{Hi}

I_{Lo} = the AQI value corresponding to BP_{Lo}

Figura 38. Formula para el calculo de ICA(Indice de calidad del aire).

Tabla 08. Subíndices específicos de contaminantes y consejos de prudenciapara obtener orientación Índice de calidad del aire.

AQI Categories (Index Values)	Ozone (ppm)		Particulate Matter (µg/m³)		Carbon Monoxide (ppm)	Sulfur Dioxide (ppb)	Nitrogen Dioxide (ppb)
	[8-hour]	[1-hour]	$PM_{2.5}$ [24-hour]	PM_{10} [24-hour]	[8-hour]	[1-hour]	[1-hour]
Good (Up to 50)	0 - 0.054 None		0 – 12.0 None	0 - 54 None	0 – 4.4 None	0 - 35 None	0 - 53 None
Moderate (51 - 100)	0.055 - 0.070 Unusually sensitive people should consider reducing prolonged or heavy outdoor exertion.		12.1 – 35.4 Unusually sensitive people should consider reducing prolonged or heavy exertion.	55 – 154	4.5 – 9.4 None	36 - 75 None	54 - 100 Unusually sensitive individuals should consider limiting prolonged exertion especially near busy roads.
Unhealthy for Sensitive Groups (101 - 150)	0.071 - 0.085 People with lung disease (such as asthma), children, older adults, people who are active outdoors (including outdoor workers), people with certain genetic variants, and people with diets limited in certain nutrients should reduce prolonged or heavy outdoor exertion.	0.125 - 0.164	35.5 – 55.4 People with heart or lung disease, older adults, children, and people of lower socioeconomic status should reduce prolonged or heavy exertion.	155 – 254	9.5 – 12.4 People with heart disease, such as angina, should limit heavy exertion and avoid sources of CO, such as heavy traffic.	76 - 185 People with asthma should consider limiting outdoor exertion.	101 - 360 People with asthma, children and older adults should limit prolonged exertion especially near busy roads.
Unhealthy (151 - 200)	0.086 - 0.105 People with lung disease (such as asthma), children, older adults, people who are active outdoors (including outdoor workers), people with certain genetic variants, and people with diets limited in certain nutrients should avoid prolonged or heavy outdoor exertion; everyone else should reduce prolonged or heavy outdoor exertion.	0.165 - 0.204	55.5 – 150.4 People with heart or lung disease, older adults, children, and people of lower socioeconomic status should avoid prolonged or heavy exertion; everyone else should reduce prolonged or heavy exertion.	255 – 354	12.5 – 15.4 People with heart disease, such as angina, should limit moderate exertion and avoid sources of CO, such as heavy traffic.	186 – 304 Children, people with asthma, or other lung diseases, should limit outdoor exertion.	361 - 649 People with asthma, children and older adults should avoid prolonged exertion near roadways; everyone else should limit prolonged exertion especially near busy roads.
Very Unhealthy (201 - 300)	0.106 - 0.200 People with lung disease (such as asthma), children, older adults, people who are active outdoors (including outdoor workers), people with certain genetic variants, and people with diets limited in certain nutrients should avoid all outdoor exertion; everyone else should reduce outdoor exertion.	0.205 - 0.404	150.5 – 250.4 People with heart or lung disease, older adults, children, and people of lower socioeconomic status should avoid all physical activity outdoors. Everyone else should avoid prolonged or heavy exertion.	355 – 424	15.5 – 30.4 People with heart disease, such as angina, should avoid exertion and sources of CO, such as heavy traffic.	305 – 604 [24-hour] Children, people with asthma, or other lung diseases should avoid outdoor exertion; everyone else should reduce outdoor exertion.	650 - 1249 People with asthma, children and older adults should avoid all outdoor exertion; everyone else should avoid prolonged exertion especially near busy roads.
Hazardous (301 - 500)	- Everyone should avoid all outdoor exertion.	0.405 - 0.604	250.5 – 500.4 Everyone should avoid all physical activity outdoors; people with heart or lung disease, older adults, children, and people of lower socioeconomic status should remain indoors and keep activity levels low.	425 – 604	30.5 – 50.4 People with heart disease, such as angina, should avoid exertion and sources of CO, such as heavy traffic; everyone else should limit heavy exertion.	605 – 1004 [24-hour] Children, people with asthma, or other lung diseases, should remain indoors; everyone else should avoid outdoor exertion.	1250 - 2049 People with asthma, children and older adults should remain indoors; everyone else should avoid all outdoor exertion.

Planificación del esquema de componentes.

Se planifica la estructura de archivos que componen las vistas, componentes y lógica del proyecto, cada uno esté compuesto por su respectiva carpeta para generar mayor organización.

Esquema general de la aplicación.

Las vistas, componentes, pipes y servicios generados se encuentran dentro de la carpeta “app”, al generar un elemento que conforme parte de la aplicación se asigna un nombre representativo.

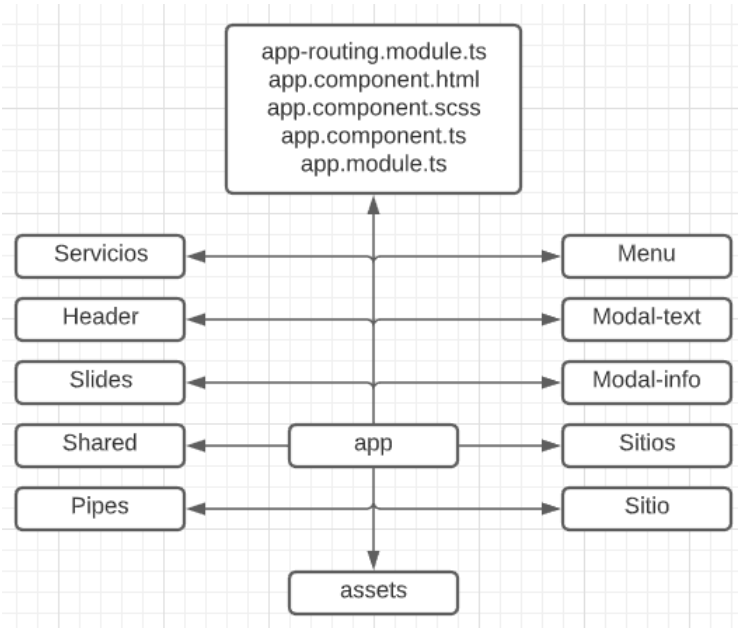

Figura 39. Esquema de componentes.

Esquema de la Vista Slides.

Esquema de archivos que conforman la vista “Slides”.

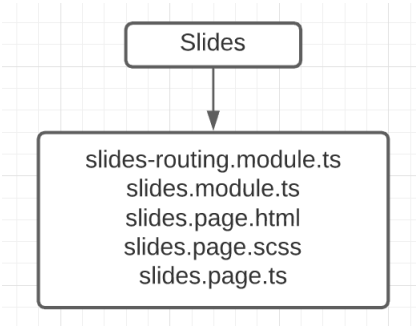

Figura 40. Esquema de archivos de la Vista “Slides”

Esquema de la Vista Sitios.

Esquema de archivos que conforman la vista “Sitios”.

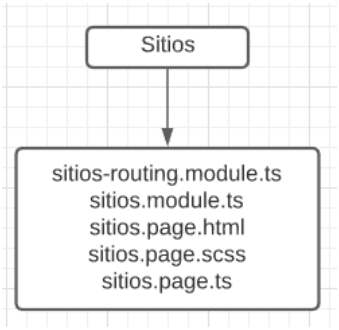

Figura 41. Esquema de archivos de la Vista “Sitios”.

Esquema de la Vista Modal-info.

Esquema de archivos que conforman la vista “modal-info”.

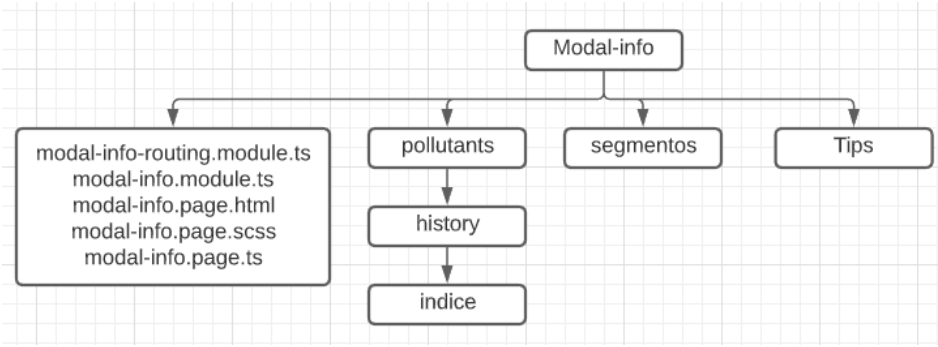

Figura 42. Esquema de archivos de la Vista “Modal-info”.

Estructura de componente pollutants.

Esquema de archivos que conforman el componente “pollutants”, forma partedel modal-info.

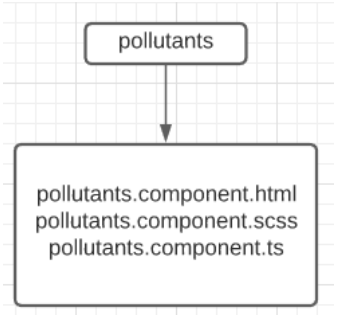

Figura 43. Esquema de archivos del componente “pollutants”.

Estructura de subcomponente indice.

Esquema de archivos que conforman el subcomponente “indice”, forma partedel componente “pollutants”.

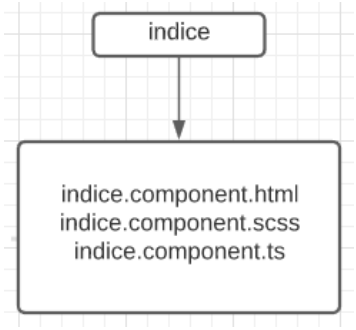

Figura 44. Esquema de archivos del subcomponente “indice”

Estructura de componente history.

Esquema de archivos que conforman el subcomponente “history”, forma partedel componente “pollutants”.

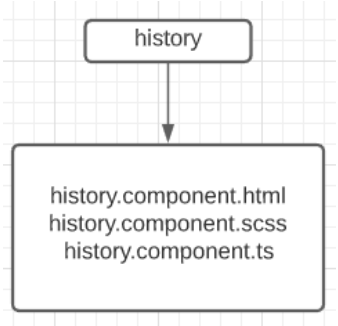

Figura 45. Esquema de archivos del subcomponente “history”

Estructura de subcomponente segmentos.

Esquema de archivos que conforman el componente “segmentos”, forma partedel modal-info.

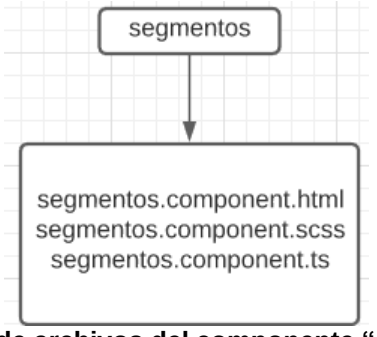

Figura 46. Esquema de archivos del componente “segmentos”.

Estructura de componente Tips.

Esquema de archivos que conforman el componente “tips”, forma parte delmodal-info.

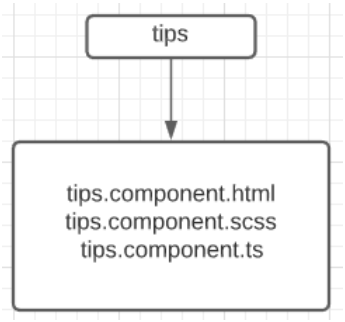

Figura 47. Esquema de archivos del componente “tips”.

Esquema de la Vista Modal-text.

Esquema de archivos que conforman la vista “modal-text”.

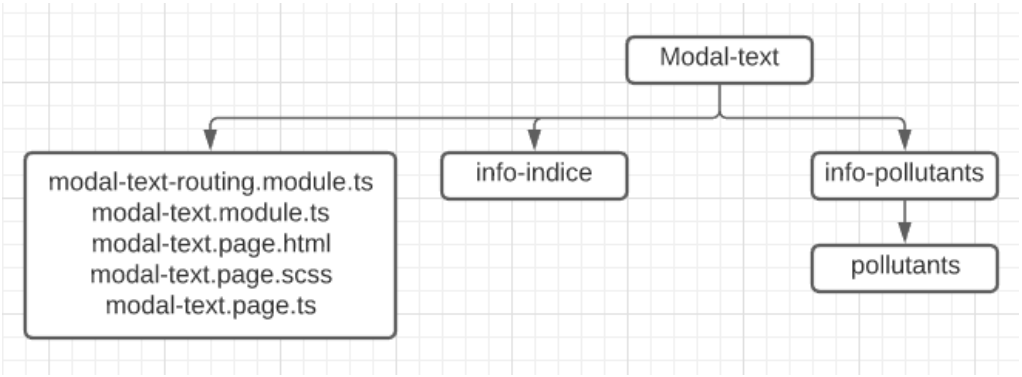

Figura 48. Esquema de archivos de la Vista “Modal-text”.

Estructura de componente info-indice.

Esquema de archivos que conforman el componente “info-indice”, forma partedel modal-text.

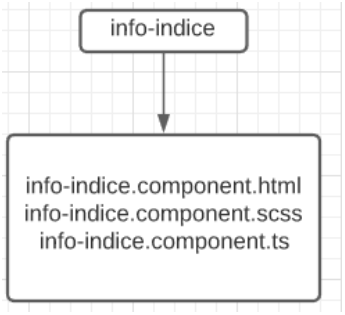

Figura 49. Esquema de archivos del componente “info-indice”.

Estructura de componente info-pollutants.

Esquema de archivos que conforman el componente “info-indice”, forma partedel modal-text.

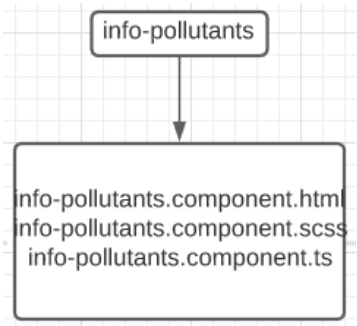

Figura 50. Esquema de archivos del componente “info-pollutants”.

Estructura de componente pollutants.

Esquema de archivos del modelo de datos que brinda para el componente “info-pollutants”.

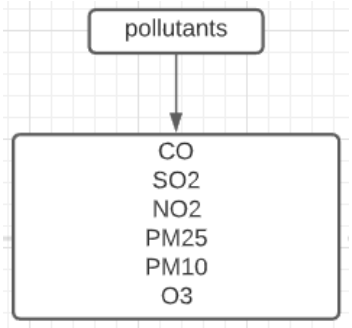

Figura 51. Esquema de modelo de datos “pollutants”.

Esquema del componente app.

Esquema de archivos que conforman el componente principal “app”.

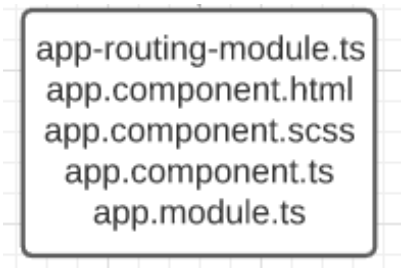

Figura 52. Esquema de archivos “app”.

Esquema del componente Header.

Esquema de archivos que conforman el componente “header”.

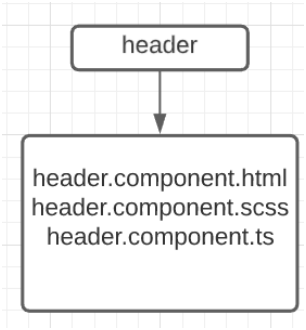

Figura 53. Estructura de carpeta “Header”.

Esquema del componente Menu.

Esquema de archivos que conforman el componente “header”.

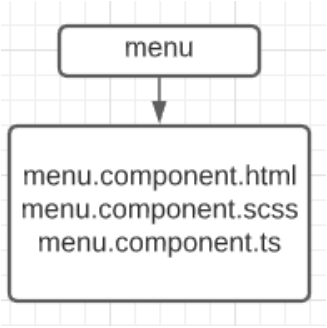

Figura 54. Estructura de carpeta “Menú”.

Esquema del componente Sitio.

Esquema de archivos que conforman el componente “sitio”.

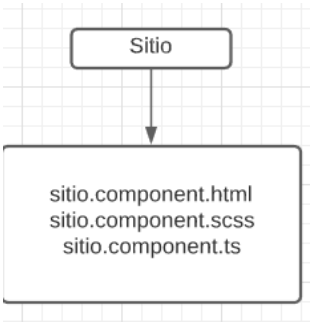

Figura 55. Estructura de carpeta “Sitio”.

Esquema del componente Shared.

Esquema de archivos que conforman el componente “Shared”.

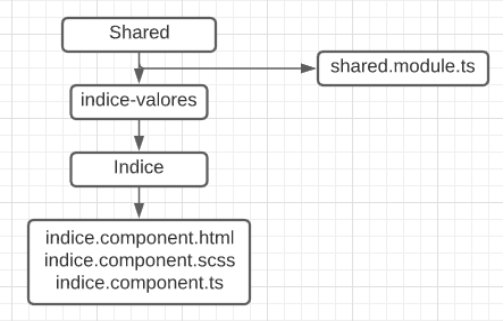

Figura 56. Estructura de carpeta “Shared”.

Esquema de Pipes.

Esquema de carpetas que conforman los pipes del proyecto, su utilidad principales el filtro de datos.

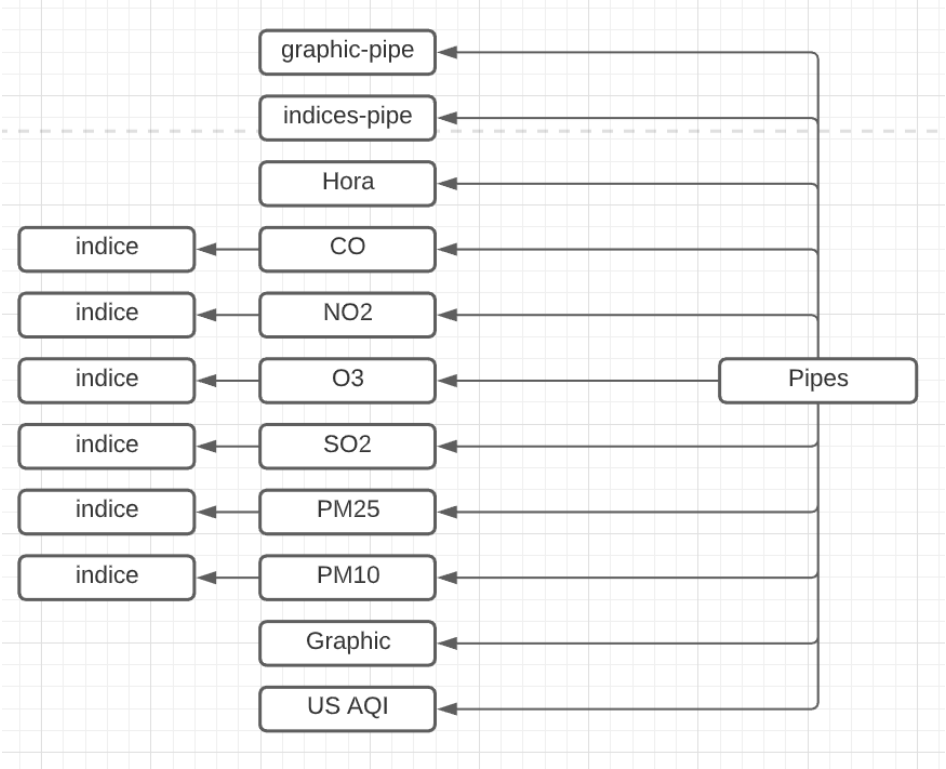

Figura 57. Estructura de carpeta “Pipes”.

Esquema de Servicios.

Esquema de carpetas que conforman los servicios necesarios para el proyecto,su utilidad principal es obtener los datos de manera local y remota.

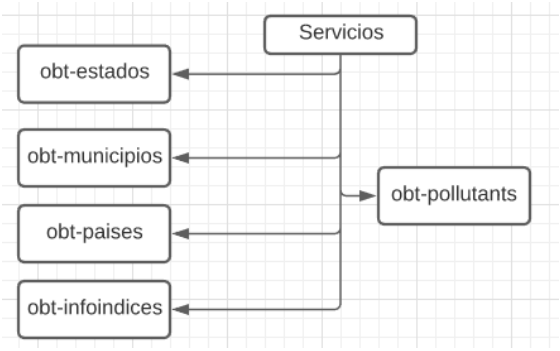

Figura 58. Estructura de carpeta “Servicios”.

Assets

Esquema de la carpeta de los recursos assets.

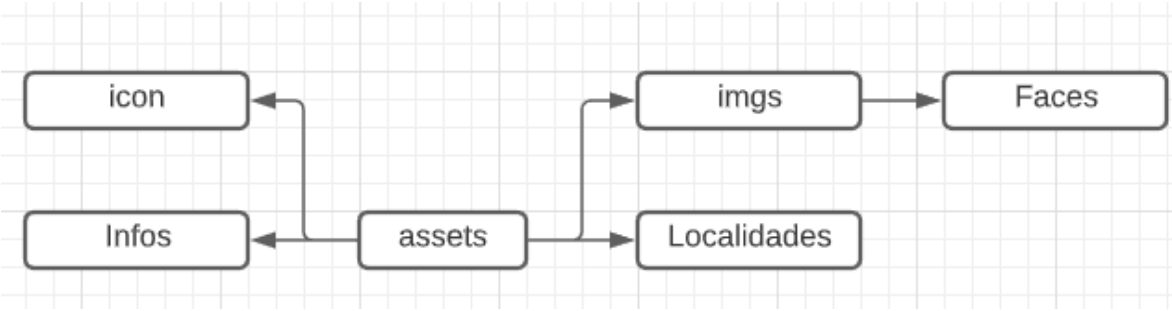

Figura 59. Estructura de carpeta “Assets”.

Prototipado.Figma

Aplicación la cual es posible diseñar interfaces de aplicaciones ya sea para web, aplicaciónmóvil o de escritorio, además es posible realizar el prototipado de proyectos colaborativamente en tiempo real, creando grupos de trabajo para colaborar con el mismo diseño.

Está basada en navegador y muy apta para la experiencia de usuario e interfaz de usuario,además es posible generar código css.

Sistema de comentarios integrados para obtener retroalimentación y tomar decisiones correctas a futuro, dentro del prototipo es posible mencionar a miembros del equipo para notificarlos sobre los cambios o la retroalimentación obtenida.

Diseño Vista Slides.

Vista de bienvenida al ser la primera vez que se ejecuta la aplicación se mostrará un conjunto de vistas que conforman la vista Slides.

Su principal función como experiencia de usuario es familiarizar al usuario con loscomponentes diseñados.

Slide 1: Muestra el componente “Sitio” y sus subcomponentes en conjunto.Slide 2:

Muestra los contaminantes criterio que se monitorean.

Slide 3: Se le pide al usuario elegir su país y estado que actualmente vive, y mostrar las estaciones de monitoreo que actualmente se encuentran disponibles en determinadas ciudades.

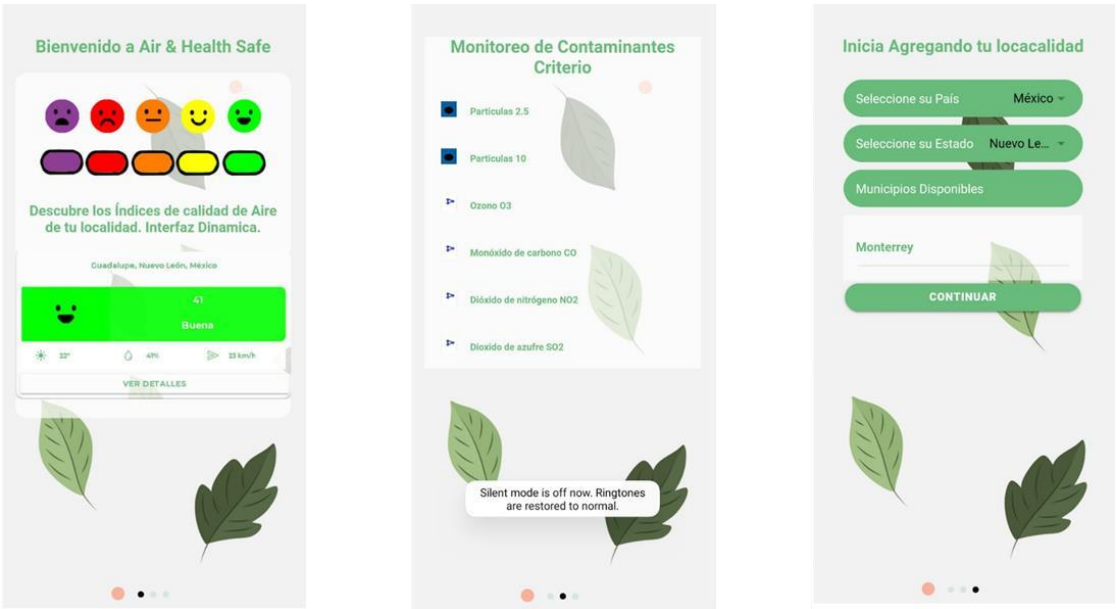

Figura 60. Slide 1, 2 y 3

Diseño Vista Sitios

Interfaz de la vista principal muestra al usuario las estaciones de monitoreo disponibles ypreferentes.

Está compuesta por varios componentes.

- Componente Sitio (Figura 61).
- Componente Shared índice (Figura 62).
- Componente Header.
- Componente Menu.

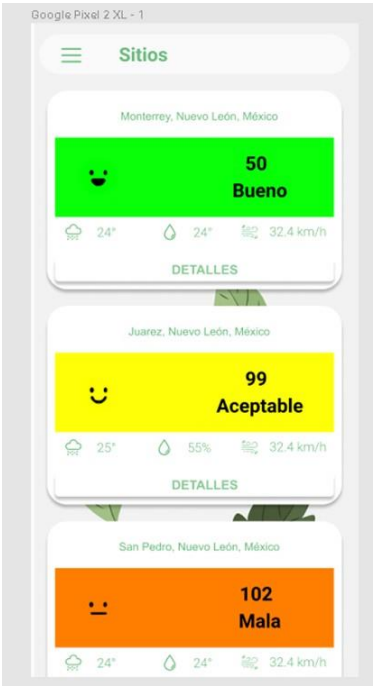

Figura 61. Diseño Vista Sitios.

Diseño componente Sitio

Componente encargado de mostrar las variables meteorológicas y el índice de calidad delaire del contaminante P.M. 2.5.

Se compone del subcomponente Shared índice (Figura 62.)

.

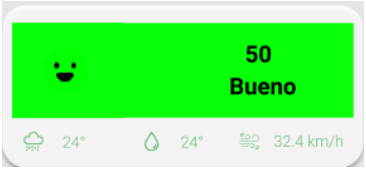

Figura 62. Diseño Shared índice.

Diseño Vista Modal-info

La interfaz de la vista Modal-info muestra información general de alguna estación de monitoreo situada en una localidad específica.

Está compuesta por varios componentes asignados para cumplir una tarea.

- Component Pollutants (Figura 64). Subcomponentes Índices (Figura 65).
- Component History (Figura 66). Subcomponents segmentos (Figura 67).
- Component Tips (Figura 68).

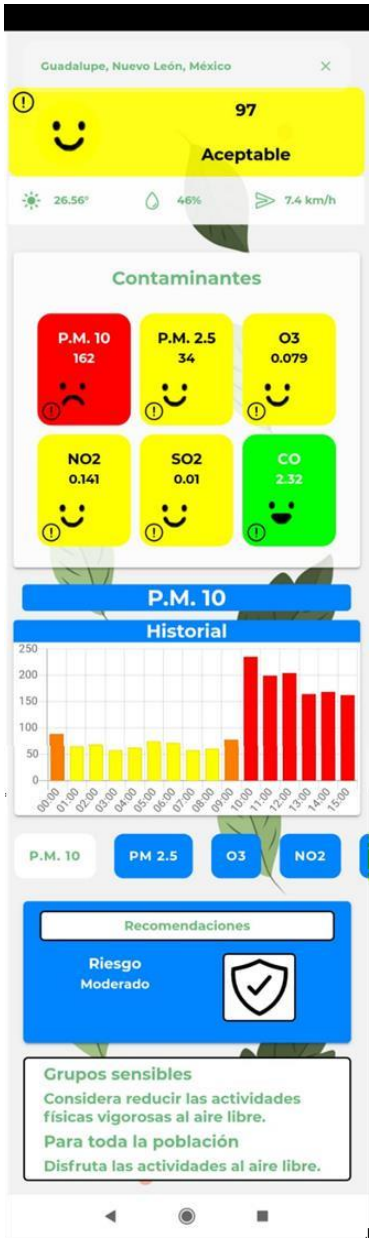

Figura 63. Diseño Vista Modal-info.

Diseño componente Pollutants

Componente encargado de generar los subcomponentes índices de calidad del aire de loscontaminantes criterio.

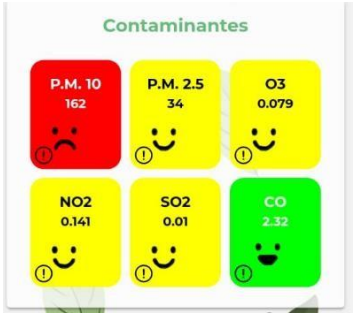

Figura 64. Diseño Componente "Pollutants".

Diseño subcomponente índice.

Subcomponente encargado de elegir el color y carita representativa del rango en el que se encuentra el nivel de concentración del contaminante criterio.

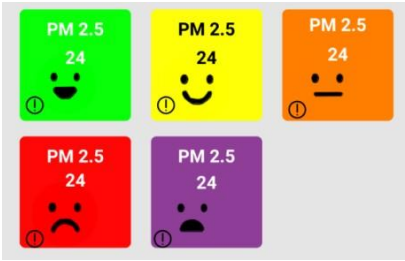

Figura 65. Diseño Subcomponente "indice".

Diseño componente history.

Componente encargado de mostrar la gráfica de un determinado contaminante, desde lahora actual hasta el comienzo del día, mostrando los registros por hora.

Figura 66. Diseño Componente "History".

Diseño subcomponente segmentos.

Subcomponente encargado de facilitar la elección de contaminante que se desea ver en lagráfica, según la elección del usuario se mostrara la gráfica en el componente “history”.

Figura 67. Diseño Subcomponente “segmentos”.

Diseño componente tips.

Componente encargado de mostrar contextualmente el nivel de riesgo y las recomendaciones, basado en el índice de la calidad del aire actual.

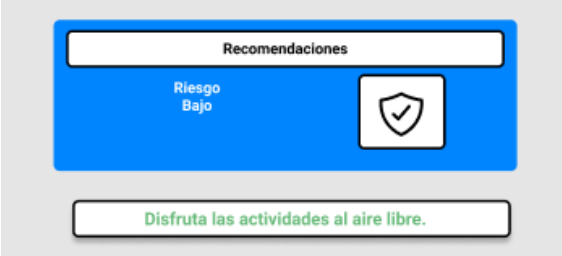

Figura 68. Diseño Componente “tips”.

Diseño Vista Modal-text

La función de este Modal es permitir conocer al usuario información acerca de los contaminantes criterio y sus índices respectivamente sea la elección del usuario, además otorga información de algunos efectos en la salud.

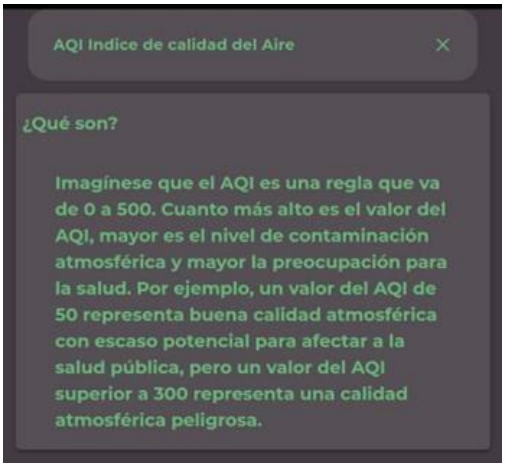

Figura 69. Diseño Vista Modal-info.

Para acceder al “Modal Información” es necesario hacer clic en el siguiente icono.

Figura 70. Diseño Icono “info”.

Diseño componente info-pollutants.

Componente específicamente encargado de mostrar la información del índice de la calidaddel aire, en el modal-info.

Figura 71. Diseño Componente “info-pollutants”.

Diseño componente Header.

Componente encargado de mostrar el título de la vista en la que se encuentra el usuarioactualmente, además de tener un icono para abrir el componente menú.

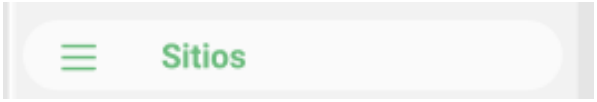

Figura 72. Diseño Componente "header".

Diseño componente Menú.

Componente encargado de mostrar las rutas en las que puede navegar el usuario, sin embargo, solo hay una en existencia.

Figura 73. Diseño Componente "menu".

Diseño componente Shared índice.

Componente encargado de seleccionar la carita, color y calidad del aire, respecto al índiceactual, se comparte para mostrar en la vista Sitios y Moda-info.

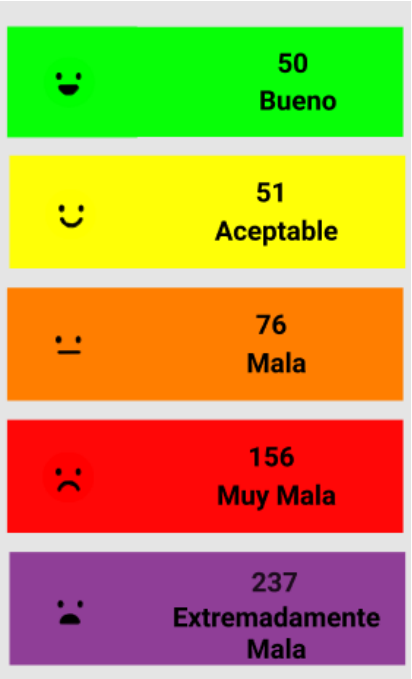

Figura 74. Diseño Componente “Shared indice”.

Desarrollo.

Los archivos que conforman cada uno de las vistas, componentes, pipes y servicios serándescrito y se utilizara la sintaxis y nombre de cada uno. (Consulte 2.5.5), solo serán mostrados los archivos que fueron afectados en la implementación de código.

Vista Slides. Módulo de

Rutas.

Ninguna modificación en el archivo.

Plantilla HTML.

Código HTML del slide 1(Figura 60), se utiliza la etiqueta “<ion-slide>” para abrir y “</ion- slide>” para cerrar la etiqueta, código HTML que se escriba dentro de estas etiquetas seránmostradas en el primer slide.

```
<ion-slide>
<div class="Titulo">
<h1>Bienvenido a Air & Health Safe</h1>
</div>
<div class="division">
<img src="../../assets/imgs/escala indice finalpmg.png" >
<p>Descubre los Índices de calidad de Aire de tu localidad.Interfaz Dinamica.</p>
<img src="../../assets/imgs/sitio.JPG" alt="">
</div>
</ion-slide>
```

Figura 75. Plantilla HTML “slide 1”.

Código HTML del slide 2(Figura 60), se utiliza la etiqueta “<ion-slide>” para abrir y “</ion- slide>” para cerrar la etiqueta, código HTML que se escriba dentro de estas etiquetas seránmostradas en el segundo slide.

```
<ion-slide>
<div class="Slide-cont">
<h1>Monitoreo de Contaminantes Criterio</h1>
<ion-item class="ion-no-
border" *ngFor="let item of items" lines="none">
<img width="40" height="40" class="img-p25"
src={{item.src}} alt="">
<h4>{{item.name}}</h4>
</ion-item>
</div>
</ion-slide>
```

Figura 76. Plantilla HTML “slide 2”.

Código HTML del slide 3(Figura 60), se utiliza la etiqueta “<ion-slide>” para abrir y “</ion- slide>” para cerrar la etiqueta, código HTML que se escriba dentro de estas etiquetas seránmostradas en el tercer slide.

```
<ion-slide>
            <div class="Titulo">
              <h1>Inicia Agregando tu locacalidad </h1>
            </div>
            <ion-footer class="ion-no-border hide" id="footer">
        <ion-grid fixed>
          <ion-row>
            <ion-col size="10" offset="1" class="Selects">
              <ion-item lines="none">
                <ion-label>Seleccione su País</ion-label>
                <ion-
select  mode="ios" placeholder="País" (ionChange)="obtest(Paisn)" [(ngModel)
]="Paisn" cancelText="Cancelar" okText="Aceptar">
                  <ion-select-
option  *ngFor="let Paises of Pais" value="{{Paises.Name}}">{{Paises.Name}}<
/ion-select-option>
                </ion-select>
              </ion-item>
              <ion-item id="Estado" lines="none" class="hide">
                <ion-label>Seleccione su Estado</ion-label>
                <ion-
select mode="ios" placeholder="Estado" (ionChange)="obtmunicipios(Paisn,Esta
don)" [(ngModel)]="Estadon" cancelText="Cancelar" okText="Aceptar">
                 <div  *ngFor="let Estado of Estados">
                   <div *ngIf="Estado.Enable">
                    <ion-select-
option  class="disponible" value="{{Estado.Entidad}}">
                      {{Estado.Name}}
                  </ion-select-option>
                </div>
                 </div>

                 </ion-select>
               </ion-item>
               <ion-item id="Municipios" lines="none" class="hide ">
                 <ion-label>Municipios Disponibles</ion-label>
               </ion-item>
                 <ion-list class="hide Disp-Municipios" id="list-m">
                   <div *ngFor="let M of Municipios">
                     <ion-item  class="Disp-Municipios" ng *ngIf="M.Enable">
                     {{M.Name}}
                     </ion-item>
                   </div>
                   </ion-list>
 <ion-button id="btn-acept" color="acept" class="btn-acept hide"
                   expand="full"
                   shape="round"
                  routerLink="/sitios"
      ><ion-label>Continuar</ion-label> </ion-button>
          </ion-col>
          </ion-row>
        </ion-grid>
    </ion-footer>
    </ion-slide>
```

Figura 77. Plantilla HTML “slide 3”.

Hoja de estilos.

```
ion-content{
--background:url("../../assets/imgs/F_Hojas.png");
background-size: auto;
}
ion-slide{
margin-bottom: 35px;
display: block; position:
relative;bottom: 0px;
}
ion-slides{
--bullet-background:rgba(105,187,123,1);
--bullet-background-active:black; position:
relative;
bottom: 0px;height:
100%;
}
```

Figura 78. Hoja de estilos de “Slides”.

Clase.

Se realizan las importaciones necesarias de los servicios a ejecutar, como lo es obtener lospaíses, estados y municipios disponibles. Además de importar “Subject” y “takeUntil”, nos permite tener el control de un observable, donde más adelante se declara, y “takeUntil” permite tomar los valores hasta que un notificador observable emita otro.

Es importante tener control en observables para evitar fugas, que causaran ralentización en la aplicación debido a su continua ejecución.

```
import { ObtPaisesService } from '../Servicios/obt-paises/obt-
paises.service';
import { ObtEstadosService } from '../Servicios/obt-estados/obt-
estados.service';
import { ObtMunicipiosService } from '../Servicios/obt-municipios/obt-
municipios.service';
import { Subject } from 'rxjs/internal/Subject';import
{ takeUntil } from 'rxjs/operators';
```

Figura 79. Clase; imports “Slides”.

Se realiza la declaración de “onDestroy$” de tal manera que $ al final significa que es un observable.

```
private onDestroy$ = new Subject<void>();
```

Figura 80. Clase; Subject “Slides”.

Declaración de servicios “obtPaisesService”, “obtEstadosService”, “ObtMunicipiosService”, es necesario declarar un variable del tipo de servicio para acceder a toda la funcionalidad de este mismo.

```
constructor(private obtpservice: ObtPaisesService, private
obteservice: ObtEstadosService,
private obtmservice: ObtMunicipiosService) { }
```

Figura 81 Clase; constructor “Slides”.

El método “ngOnInit” es propio del framework Ionic, pertenece al ciclo de vida de este mismo, su función es realizar acciones al iniciar el componente, en el código se requiere elservicio de obtener países utilizando observables, y un manejador de observables para sudestrucción.

```
ngOnInit() {
this.obtpservice.obtpais().p
ata) =>this.Pais = data);
}
```

Figura 82. Clase; ngOnInit “Slides”.

El método “ngOnDestroy” es propio del framework Ionic, pertenece al ciclo de vida de estemismo, su función es realizar acciones al destruirse el componente, en el código se finalizanlos observables ya abiertos en “ngOnInit”.

```
ngOnDestroy( this.onDestroy$.next();
this.onDestroy$.complete();
}
```

Figura 83 Clase; ngOnDestroy “Slides”.

Método que se ejecuta al mostrar el ultimo slide, muestra el componente “footer”.

```
endslide(){
document.getElementById("footer").style.display="block";
}
```

Figura 84. Clase; Método endslide “Slides”.

Método ejecutado por el selector de país, su parámetro recibido es el país seleccionado, por consiguiente, trae los estados pertenecientes a este mismo, en caso de no seleccionarninguno el valor es indefinido y no sucede nada.

```
obtest(
if(Paisselec
}else{ this.obteservice.obtE(Paisselect).
scribe((data) => this.Estados = data);
document.getElementById("Estado").style.display= "block"
}
```

Figura 85. Clase; Método obtest “Slides”.

Método ejecutado por selector de estados, su parámetro recibido es el estado seleccionado,por consiguiente, trae los municipios pertenecientes a este mismo, en caso de no seleccionar ninguno el valor es indefinido y no sucede nada.

```
obtmu
if(Paiss this.obtmservice.
estroy$)).subscribe((data)=> th
document.getElementById("Municipio
document.getElementById("list-m").style.di
document.getElementById("btn-acept").style.display
}
}
```

Figura 86. Clase; Método obtmunicipios “Slides”.

Módulo.

Se realizan las importaciones necesarias para hacer funcionar el componente y hacer usode funciones externas.

```
import { SlidesPage } from './slides.page';
import { ObtPaisesService } from '../Servicios/obt-paises/obt-
paises.service';
import { ObtEstadosService } from '../Servicios/obt-estados/obt-
estados.service';
import { ObtMunicipiosService } from '../Servicios/obt-municipios/obt-
municipios.service';
import { SitioComponent } from '../sitio/sitio.component';import
{ SharedModule } from '../Shared/shared.module';
```

Figura 87. Módulo; imports “Slides”.

En el módulo fueron requeridos componentes y servicios externos, los cuales sondeclarados en las propiedades correspondientes.

```
@NgMo
imports: [
CommonModule,
FormsModule,
IonicModule,
SlidesPageRoutingModule,
SharedModule
],
declarations: [SlidesPage,SitioComponent],
providers: [ObtPaisesService, ObtEstadosService, ObtMunicipiosServ
})
```

Figura 88. Módulo; Directiva @NgModule “Slides”.

Vista Sitios Módulo de

Rutas.

Ninguna modificación en el archivo.

Plantilla HTML.

Componente “app-header” usado al principio del código HTML.

```
<app-header sitio="Sitios"></app-header>
```

Figura 89. Plantilla HTML; app-header “Sitios”.

Para la generación de componente sitio se realiza la directiva “ngFor” donde se recorre elarreglo que guarda la información de una determinada ubicación.

```
<div *ngIf="zona">
<ion-
col sizeXs="12" sizeSm="12" sizeLg="6" sizeMd="6" sizeXl="4" *ngFor="let u of Zona">
<div *ngFor="let h of u| city:Hour" >
<app-sitio Temp="{{h.Temp}}" Humedad="{{h.Humedad}}"
Viento="{{h.Viento}}"
[Values-P]="h"
AQI="{{AQI}}">
                </app-sitio>
              </div>
            </ion-col>
             </div>
```

Figura 90. Plantilla HTML; app-sitio “Sitios”.

Hoja de estilos.

```
ion-content{
--background:url("../../assets/imgs/F_Hojas.png");
}
#loading{
position: absolute;width:
100%;
top: 45%; ion-
spinner{
position: relative;left: 45%;
--color:#69bb7b;
}}
```

Figura 91. Hoja de estilo de “Sitios”.

Clase.

Se realizan las importaciones necesarias para hacer funcionar la vista.

```
import { S
pollut.service';
import { finalize, takeUntil } from 'rxjs
import { Subject } from 'rxjs/internal/Subject'; import {
UsaqiPipe } from '../Pipes/US AQI/usaqi.pipe';
```

Figura 92. Clase; imports “Sitios”.

Se realiza la declaración de “onDestroy$” de tal manera que $ al final significa que es un observable.

```
private onDestroy$ = new Subject<void>();
```

Figura 93 Clase; Subject “Sitios”.

Declaración de variables globales

Zonas: Representa el estado para mostrar el componente “sitio”, en caso de ser “true” se muestran el componente sitio, en este caso es cuando termina la ejecución de petición a firebase de datos.

Loading: Representa el estado para mostrar el componente de carga, en caso de ser “true”se muestra la animación de típica de cargando datos.

Zona []: Arreglo donde se guardan los datos recibidos de firebase con todos los atributos.AQI:

Variable donde se guarda el índice de calidad calculado del Pipe “UsaqiPipe”.

Hour: Variable donde se guarda la hora actual en formato 24 horas, siendo menor a 2 dígitosla hora solo se obtiene uno, se concatena “:00”: para cumplir con el formato de hora guardado en la base de datos.

```
zona=fal
loading=true;Zona =
[]; AQI;
Hour=new Date().getHours()+":00";
```

Figura 94. Clase; Subject “Sitios”.

Declaración de servicio “ServicePollutService” y el pipe “UsaqiP”, es necesario declarar unvariable del tipo de servicio y pipe para acceder a toda la funcionalidad de este mismo.

```
constructor(private pollutants:ServicePollutService, private usaqi: UsaqiPipe) { }
```

Figura 95. Clase; constructor “Sitios”.

Accedemos a la función “Pollutants” del servicio “ServicePollutService”, para realizar la petición a la base de datos en Firebase, utilizando observables y tener el control de flujo dedatos sobre esta función.

Utilizamos un for dentro de la función para recorrer nuestro objeto adquirido de firebase, deesta manera es posible acceder a los valores que son requeridos, por consiguiente, son guardados en el arreglo “Zona” utilizando push, al término del for es utilizado el cambio devalor para las variables zona, loading.

Se accede a la función “transform” del pipe “UsaqiP” para determinar el valor del índice decalidad del aire de la hora actual, como parámetros son tipo arreglo y string, se envía el arreglo Zona y la Hora actual.

```
ngOnInit() { this.pollutants.Pollutants().pipe(takeUntil(this.onDestroy$),finalize(()
       =>{
           })).subscribe(action=>{
           let f=[];
           f.push(action);
           for(let i of f){
           this.Zona.push(i.payload.val())
           }
this.zona=true; this.loading=false;
this.AQI=  this.usaqi.transform(this.Zona,this.Hour);
}                        );
}
```

Figura 96. Clase; ngOnInit “Sitios”.

El método “ngOnDestroy” es propio del framework Ionic, pertence al ciclo de vida de este mismo, su función es realizar acciones al destruirse el componente, en el código se finalizanlos observables ya abiertos en “ngOnInit”.

```
ngOnDestroy(): void {
this.onDestroy$.next();
this.onDestroy$.complete();
}
}
```

Figura 97. Clase; ngOnDestroy “Sitios”.

Módulo

Se realizan las importaciones necesarias para hacer funcionar el componente y hacer usode funciones externas.

```
import { SitiosPage } from './sitios.page';
import { HeaderComponent } from '../header/header.component'; import {
ModalInfoPage } from '../modal-info/modal-info.page';import { ModalTextPage } from
'../modal-text/modal-text.page'
import { ModalInfoPageModule } from '../modal-info/modal-info.module';import {
SitioComponent } from '../sitio/sitio.component';
import { SharedModule } from '../Shared/shared.module'import { CityPipe }
from '../Pipes/Hora/city.pipe' import { UsaqiPipe} from '../Pipes/US
AQI/usaqi.pipe';
```

Figura 98. Módulo; imports “Sitios”.

En el módulo fueron requeridos componentes y servicios externos, los cuales sondeclarados en las propiedades correspondientes.

```
@NgM
entryC
ModalInfo ModalTextPage
],
imports: [
CommonModule,
FormsModule,
IonicModule,
SitiosPageRoutingModule,
ModalInfoPageModule, SharedModule
],
declarations: [SitiosPage,HeaderComponent,SitioComponent,CityPipproviders:[HTTP,UsaqiPipe]
})
```

Figura 99. Módulo; Directiva @NgModule “Sitio”.

Componente Sitio

Plantilla HTML.

Interpolación para mostrar la locación donde se están recibiendo los datos de Firebase.

```
<ion-card-subtitle > {{Values.Location}}</ion-card-subtitle>
```

Figura 100. Plantilla HTML; Location “Sitio”.

Se utiliza la directiva ngSwitch, en base al índice de calidad de aire, ya calculado en el pipe“UsaqiP” se comprueba cual caso es verdadero para asignar la propiedad para el color correcto en su correspondiente rango de calidad del aire.

```
<i
<i c="gre
Humedad
<= 50">
</indice>
<indice Class
c="yellow_face"
Humedad="{{HumedadViento="{{Viento}}"
</indice>
<indice Class-c="Mala" Ca
c="orange_face"           Temp="{{Te
Humedad="{{Humedad}}" AQI="{{
Viento="{{Viento}}"  *ngSwitchCa
</indice>
<indice  Class-c="Muy-Mala"  Calidad-c
c="red_face"          Temp="{{Temp}}"
Humedad="{{Humedad}}" AQI="{{AQI}}"
Viento="{{Viento}}" *ngSwitchCase="AQI > 15
</indice>
<indice  Class-c="Ex-Mala"  Calidad-c="Extremadam
c="purple_face"           Temp="{{Temp}}"
Humedad="{{Humedad}}" AQI="{{AQI}}" Viento="{{Viento}}"
*ngSwitchCase="AQI  > 200">
</indice>
<indice  *ngSwitchDefault  Class-c="Bueno"  Calidad-c="Buena"
c="green_face"          Temp="22"
Humedad="32"  Viento="32"  AQI="{{AQI}}">
</indice>
<ion-button  class="button-
details"  (click)="abrirModal(Values,Temp,Humedad,Viento,AQI)">Ver  Detallion-
button>
</ion-col>
```

Figura 101. Plantilla HTML; Switch Color “Sitio”.

Hoja de estilos.

Se importa tipografía “Montserrat” estilo “sans-serif” el cual se utiliza en el tipo de letra.

```
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@700&fa
mily=Roboto:wght@700&display=swap');
$font: 'Montserrat', 'sans-serif';ion-
card{
--background: rgba(255,255,255,.8) ;border-
radius: 15px;
text-align: center;
}

ion-card-subtitle{ font-
size: 14px; font-weight:
600;color: #69BB7B;
font-family: $font;
}

ion-icon{
font-size: 24px;
color:gray;

}
.center{ display:
flex;
justify-content: center;
}

.button-details{
--background:rgba(255,255,255,.1);color:#69BB7B;
font-weight: 800;
width: 100%;
font-family: $font;
}
```

Figura 102. Hoja de estilo “Sitio”.

Clase.

Se realiza las importaciones para el controlador de modales y el “ModalInfo”.

```
import { ModalController } from '@ionic/angular';
import { ModalInfoPage } from '../modal-info/modal-info.page';
```

Figura 103. Clase; imports “Sitio”.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

Humedad: Propiedad de tipo string para el valor Humedad actual.Viento:

Propiedad de tipo string para el valor Viento actual.

Temp: Propiedad de tipo string para el valor de Temperatura actual.

Values-P: Propiedad de tipo arreglo, guarda los valores correspondientes de cada hora,contaminantes y locación.

AQI: Propiedad de tipo cualquiera, muestra el índice de la calidad del aire actual dedeterminada locación.

```
@Input("H

@Input("Viento") Viento:string;
@Input("Temp") Temp:string;
@Input("Values-P") Values:any;
@Input("AQI") AQI:any;
```

Figura 104. Clase; Inputs “Sitio”.

Función asíncrona donde se controla el “modalinfo”, se reciben como parámetros:Values:

De tipo arreglo, se pasa la propiedad “Values-P”.

T: De tipo string, se pasa la propiedad “Temperatura”.H: De

tipo string, se pasa la propiedad “Humedad”.

Viento: De Tipo string, se pasa la propiedad “Viento”.AQI: De

tipo string, se pasa la propiedad “AQI”.

El componente a mostrar es “ModalInfoPage” pasando sus propiedades de la mismapágina.

Al final de la función se presenta el modal.

```
async abrirModal(Values:[],T,H,Viento,AQI){ const modal =
await this.modalCtrl.create({
component:ModalInfoPage,componentProps:{
Values:Values, Viento:Viento,
H:H,
T:T, AQI:AQI
}
});
await modal.present();
}
```

Figura 105. Clase; Función AbrirModal “Sitio”.

Vista Modal Info. Módulo

de Rutas.

Ninguna modificación en el archivo.

Plantilla HTML.

Interpolación para mostrar la locación en el título del Header..

```
<ion-title slot="start">{{Values.Location}}</ion-title>
```

Figura 106. Plantilla HTML; ion-title “Modal-info”.

Botón que ejecuta una función para salir del modal.

```
<ion-button fill="clear" class="Cerrar-
btn" slot="end" (click)="salirsinArgumentos()"><ion-icon name="close-
outline" color="acept"></ion-icon></ion-button>
```

Figura 107. Plantilla HTML; ion-button “Modal-info”.

Se utiliza la directiva ngSwitch, en base al índice de calidad de aire, ya calculado en el pipe“UsaqiP” se comprueba cual caso es verdadero para asignar la propiedad para el color correcto en su correspondiente rango de calidad del aire.

```
<div [ngSwitch]="tru
</div>
```

Figura 108. Plantilla HTML; Directiva ngSwitch “Modal-info”.

Cada caso es una condicional para saber el rango en el que se encuentra la calidad delaire, se selecciona su correspondiente estado.

```
<indice C
c="green_face"          Temp="{{T
Humedad="{{H}}" Viento="{{Viento}}" AQI
">
</indice>
```

Figura 109. Plantilla HTML; Casos de Switch “Modal-info”.

Componentes que conforman el modal-info, pasando sus respectivas propiedades.

```
<pollutants [Values]="Values" ></pollutants>
<app-segmentos city="{{Values.Location}}"></app-segmentos>
<app-tips AQI="{{AQI}}"></app-tips>
```

Figura 110. Plantilla HTML; Componente utilizados“Modal-info”.

Hoja de estilos.

Se importa tipografía “Montserrat” estilo “sans-serif” el cual se utiliza en el tipo de letra.

```
@im mily=R

$font: 'Montsion-
content{
--background:ur background-
attachme
}

.header{
top: 10px;
}

ion-toolbar{
--background: rgba(255,255, 255,.3);border-
radius: 15px;
width: 90%;
left: 5%; ion-
title{
color:#69bb7b; font-size:
12px;
font-family: $font;
}}
```

Figura 111. Hoja de estilo “Modal-info”.

Se realiza un query para obtener el tamaño de pantalla del dispositivo en base a esto, siendo el caso de ser igual o menor a 280 pixeles el ancho de pantalla se cambió el estilo en las siguientes clases mostradas en la figura 114.

```
@media only screen and(max-width:280px){ion-

toolbar{

ion-title{
font-size: 10px;
}
}
}
```

Figura 112. Hoja de estilo; Header responsivo “Modal-info”.

Clase.

Importación del controlador de modales.

```
import { ModalController } from '@ionic/angular';
```

Figura 113. Clase; imports “Modal-info”.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

Values: Propiedad de tipo arreglo, guarda los valores correspondientes de cada hora,contaminantes y locación.

T: Propiedad de tipo cualquiera, guarda la temperatura.H:

Propiedad de tipo cualquiera, guarda la Humedad.

Viento: Propiedad de tipo cualquiera, guarda el viento.

AQI: Propiedad de tipo cualquiera, guarda el índice de calidad del aire actual.

```
@Input() V @Input() T;
@Input() H; @Input()
Viento;@Input() AQI;
```

Figura 114. Clase; Inputs “Modal-info”.

Función para salir del modal sin ningún argumento a enviar a “Sitios”.

```
salirsinArgumentos(){
this.modalCtrl.dismiss();
}
```

Figura 115. Clase; Función Salir sin argumentos “Modal-info”.

Modulo.

Se realizan las importaciones necesarias para hacer funcionar el componente y hacer usode funciones externas.

```
//Paginas
import { ModalInfoPage } from './modal-info.page';
import { ModalTextPage } from '../modal-text/modal-text.page';
//Modulos
import { SharedModule } from '../Shared/shared.module';
import { IndicesPipeModule } from '../Pipes/indices-pipe/indices-
pipe.module';
import { GraphicPipeModule } from '../Pipes/graphic-pipe/graphic-
pipe.module';
//Componentes
import { PollutantsComponent } from './pollutants/pollutants.component'; import
{ IndiceComponent } from './pollutants/indice/indice.component'; import {
HistoryComponent } from './pollutants/history/history.component';import {
SegmentosComponent } from './segmentos/segmentos.component'; import {
TipsComponent } from './Tips/tips.component';
```

Figura 116. Módulo; imports “Modal-info”.

En el módulo fueron requeridos componentes y servicios externos, los cuales sondeclarados en las propiedades correspondientes.

```
@NgModule({
imports: [
CommonModule,
FormsModule,
IonicModule,
ModalInfoPageRoutingModule,
SharedModule, IndicesPipeModule,
GraphicPipeModule

],
declarations: [ModalInfoPage,PollutantsComponent,IndiceComponent,HistoryCo
mponent,SegmentosComponent,TipsComponent,ModalTextPage],
providers:[]
})
```

Figura 117. Módulo; Directiva NgModule “Modal-info”.

Componente Pollutants.

Plantilla HTML.

Componente a base de “ion-card” y “ion-grid”, encierran en “ion-col” de tamaño 4, dejando3 subcomponentes índice en cada fila.

Se repite 6 veces con su respectivo contaminante e información.

```
<ion-c
<a class="i circle-
outline">

</a>
<indice-
p Color="{{Values.PM10|ipm10}}" Pollutants="P.M. 10" Valor="
</indice-p>
</ion-col>
```

Figura 118. Plantilla HTML; Componente base indice “Pollutants”.

Hoja de estilos.

Estilo asignado para el componente card.

```
ion-ca
margin-top: 25
--background:rgba(255,2ion-
card-title{

font-family:   $font;  text-align:
center;color: #69bb7b;
}}
```

Figura 119. Hoja de estilo; ion-card “Pollutants”.

Estilo del icono información pollutant (Figura 72).

```
.info{
ion-icon{ color: black;
font-size: 24px; position:
absolute;z-index: 4;
left: 8px; bottom:8px;
cursor: pointer;
}}
```

Figura 120. Hoja de estilo; icono-info “Pollutants”.

Clase.

Importación del controlador de modales y el modal a abrir.

```
import { ModalController } from '@ionic/angular';
import { ModalTextPage } from 'src/app/modal-text/modal-text.page';
```

Figura 121. Clase; imports “Pollutants”.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

Values: Propiedad de tipo cualquiera, guarda los valores como hora, contaminantes criterio, variables meteorológicas y localidad.

```
@Input("Values") Values:any;
```

Figura 122. Clase; Inputs “Pollutants”.

Función abrirModal(p,t){}

Función asíncrona, recibe dos parámetros tipo cualquiera

p: Si es un contaminante o el índice de calidad del aire se decide que segmento se elige.t:

Selección del título de ventana del Modal text.

Variable title se asigna el título que tendrá la ventana del Modal text.

El switch es el encargado de seleccionar el título de la ventana según el número recibidodel parámetro “t”.

```
let


t brea
case t==2
title ="P.break; case t==3:
title = "O3"break;
case t==4: title = "NO2"break;
case t==5: title = "SO2"break;
case t==6: title = "CO"break;
default:break;
}
```

Figura 123. Clase; Función abrirModal Switch “Pollutants”.

Creación de nuevo modal, pasando las propiedades respectivas.

Se selecciona el componente para mostrar, siendo Modal text, y al final se presenta elmodal.

```
const
compon componentProps:{
Segment:p,Title:title
}
});
await modal.present();
```

Figura 124. Clase; Función abrirModal Creación de modal “Pollutants”.

Subcomponente índice.

Plantilla HTML.

La selección de clase para el estilo del subcomponente es previamente seleccionada a partir del componente “pollutants” y en el pipe respectivo del contaminante criterio se selecciona el estado que sería la clase del subcomponente.

Se hacen dos interpolaciones, haciendo el subcomponente dinámico en su creación.Color:

Selección de la clase de estilo para el subcomponente.

Pollutants: Nombre del contaminante criterio.

Valor: Valor del índice del contaminante criterio de la hora actual.

```
<div clas
<h2>{{Pollutants}}</h
<p >
{{Valor}}
</p>
</div>
```

Figura 125. Plantilla HTML “indice”.

Hoja de estilos.

Estilo general de todas las clases, exceptuando el color de fondo del subcomponente.

```
.Green,.Yellow,.Orange,.Red,.Purple{position:
relative;
padding: 15px 0; padding-bottom:
60px;width: 100%;
border-radius: 15px;z-index: 0;
}
```

Figura 126. Hoja de estilo; Clase Estilo general “indice”.

Se asigna su respectivo color de fondo en formato rgba siguiendo la tabla 04.

```
.Gre
back
}
.Yellow{
background-color: rp,h2{
color: black;}}
.Orange{
background-color: rgba(255,126,0,1);
}
.Red{
background-color: rgba(255,0,0,1);
}
.Purple{
background-color: rgba(143,63,151,1);
}
```

Figura 127. Hoja de estilo; Clase Colores “indice”.

El uso de pseudo clases se utiliza para colocar las caritas representativas respectivas enuna posición especifica.

```
.Green::before,.Yellow::before,.Orange::before,.Red::before,.Purple::before{
position: absolute;
content:    "";  width:
100%;height:60%;
background-size: contain; background-
repeat: no-repeat;z-index: -1;
bottom: 0%;
}
```

Figura 128. Hoja de estilo; PseudoClase before Estilo general “indice”.

Por separado se hacen las pseudo clases para asignar la respectiva imagen según el colorde la clase.

```
.Green::before{
background-image: url("../../../../assets/imgs/faces/green_face.png");
}
```

Figura 129. Hoja de estilo; PseudoClase before bk-image “indice”.

Diseño responsivo mediante query, se hace una consulta al tamaño de ancho de pantalla del dispositivo, para asignar el estilo adecuado, se realiza la adaptación de letra al hacerlamás pequeña para dispositivos más pequeños, evitando saturar el subcomponente y evitardesbordamiento.

```
@media only screen and(max-width:320px){

h2{
              font-size: 14px;
}
p{
              font-size: 12px;
}
}


@media only screen and(max-width:280px){

h2{
            font-size: 14px;
}
p{
            font-size: 12px;
     }
     }
```

Figura 130. Hoja de estilo; Responsive “indice”.

Clase.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

Color: Propiedad de tipo string, guarda el estado de la calidad del aire, su uso es asignar laclase de estilo al subcomponente.

Pollutants: Propiedad de tipo string, guarda el nombre del contaminante criterio, su uso esmostrar el nombre del contaminante criterio en específico.

Valor: Propiedad de tipo string, guarda el valor del índice del contaminante criterio, su usoes mostrar el valor de la hora actual de un contaminante criterio en específico.

```
@Input("Color") Color:string;
@Input("Pollutants") Pollutants:string;
@Input("Valor") Valor:string;
```

Figura 131. Clase; Inputs “indice”.

Subcomponente history.

Plantilla HTML.

Componente con título siendo “P” el nombre del contaminante criterio, el componentehistory está en base a un “ion-card” y una etiqueta “canvas” para dibujar la gráfica.

```
<h2>{{P}}
<ion-card>
<ion-card-title>
Historial</ion-card-title>
<canvas id="canvas">{{chart}}</canvas>
</ion-card>
```

Figura 132. Plantilla HTML “history”.

Hoja de estilos.

Estilos aplicados para las etiquetas HTML y componente UI de ionic.

```
h2{
            position:  relative;
            text-align:  center;
            color: white;
            background-color: #0085FF;
            width: 90%;
            left: 5%;
            border-radius: 5px;
            font-size: 24px;
            font-family: $font;
}

ion-card{
ion-card-title{
font-family: $font;text-align:
center;
background-color: #0085FF;color: white;

}
}
```

Figura 133. Hoja de estilo “history”.

Clase.

Importación de Chartjs necesaria para dibujar una gráfica.

```
import { Chart } from 'chart.js';
```

Figura 134. Clase; import Chartjs “history”.

Importación del Servicio “ServicePollutService”, necesario para realizar la petición aFirebase.

```
import { ServicePollutService } from 'src/app/Servicios/obt-
pollutants/service-pollut.service';
```

Figura 135. Clase; import Service Pollutants “history”.

Importaciones de rxjs, necesarias para el control de observables.

```
import { Subject } from 'rxjs/internal/Subject'; import {
finalize, takeUntil } from 'rxjs/operators';
```

Figura 136. Clase; import rxjs“history”.

Importaciones de Pipes, necesarios para realizar filtrado de datos obtenidos de Firebase,respectivamente cumplen con una tarea en específico.

```
import
import { PM10Pipe
import { SO2Pipe } from 'src/
import { PM25Pipe } from 'src/app/Pipes/
import { O3Pipe } from 'src/app/Pipes/O3/o3.pipe'; import {
CoPipe } from 'src/app/Pipes/CO/co.pipe'; import { No2Pipe
} from 'src/app/Pipes/NO2/no2.pipe';
```

Figura 137. Clase; import Pipes“history”.

Variable que maneja los observables de tipo Subject list.

```
private onDestroy$ = new Subject<void>();
```

Figura 138. Clase; onDestroy$ “history”.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

P: Propiedad de tipo string, guarda el nombre del contaminante criterio a mostrar en lagráfica.

City:

```
@Input("P") P:string;
@Input("city") city:string;
```

Figura 139. Clase; Inputs “history”.

Array vacío necesario para guardar los datos de la base de datos en Firebase.

```
Datos =[];
```

Figura 140. Clase; Array “history”.

Constantes declaradas necesarias para obtener la fecha actual.

```
Hora = new
Month = new Date().getMonth(); Day = new
Date().getDate(); Year = new
Date().getFullYear();
```

Figura 141. Clase; Constantes Fecha “history”.

Declaración se “ServicePollutService”, GrapPipe y los Pipes de los 6 contaminantes criterio.

```
cons

private P
private S02: SO2P private PM25: PM25Pipe,
private O3: O3Pipe, private CO:CoPipe,
private NO2: No2Pipe
) { }
```

Figura 142. Clase; Constructor “history”.

Función que forma parte del ciclo de vida del componente Ionic, es ejecutada según se detectan cambios en el componente, y se ejecuta la función “CHECK ()”.

```
ngOnChanges(changes): void {
this.CHECK();
}
```

Figura 143. Clase; ngOnChanges “history”.

Función que forma parte del ciclo de vida del componente Ionic, es ejecutada al iniciar elcomponente, se ejecuta la función “ver ()”.

```
ngOnInit() {
this.ver();
}
```

Figura 144. Clase; ngOnInit “history”.

Función CHECK (), se encarga de revisar si hay datos dentro del array Datos, siendo falsoeste caso, se ejecuta un switch tomando como parámetro el nombre del contaminante, de este modo se ejecutan los pipes correspondientes.

El segundo if condiciona si hay una gráfica ya dibujado, siendo el caso falso se dibuja una,siendo el caso verdadero, se ejecuta la función “removeData ()”.

```
CHECK(){
if(this.Datos.length==0){} else{
switch (this.P) {case "P.M.
10":
this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'PM10');this.Values =
this.GrapPipe.resv;
this.Colors= this.PM10.transform(this.Values)break;
case "P.M. 2.5":
this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'PM25');this.Values =
this.GrapPipe.resv;
this.Colors= this.PM25.transform(this.Values)break;
case "O3":
this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'O3');this.Values =
this.GrapPipe.resv;
this.Colors= this.O3.transform(this.Values)break;
case "NO2":
this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'NO2');this.Values =
this.GrapPipe.resv;
this.Colors= this.NO2.transform(this.Values)break;
case "SO2":
this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'SO2'
        );
                    this.Values = this.GrapPipe.resv;
                    this.Colors= this.S02.transform(this.Values)
                    break;
                    case "CO":
                    this.HoraD = this.GrapPipe.transform(this.Datos,this.Hora,'CO
        ');
                    this.Values = this.GrapPipe.resv;
                    this.Colors= this.CO.transform(this.Values)
                    break;
           default:
           break;


}
if(this.chart){ this.removeData(this.chart);
}else{ this.createChart();
}
```

Figura 145. Clase; Función CHECK “history”.

Función que recibe un parámetro de tipo chart, encargada de resetear a vacío laspropiedades de la gráfica chart, de esta manera será posible pintar diferentes gráficos.

Chart.update(), método necesario para actualizar la gráfica.

Ejecución del método addData (), necesario para añadir la nueva información.

```
remov chart.data.l
chart.data.datasets.f
dataset.data=[];
dataset.backgroundColor=[];
});
chart.update(); this.addData(this.chart,this.HoraD,this.Values,this.Colors);
}
```

Figura 146. Clase; Función removeData “history”.

Función que recibe como parámetros de tipo chart, label, data y colors, respectivamente, encargada de asignar valores a las propiedades de la gráfica, de esta manera pintara los datos a mostrar.

Chart.update(), método necesario para actualizar la gráfica.

```
addD chart.data.
chart.data.datasets

dataset.data =data; dataset.backgroundColor
=(colors);

});
chart.update();
}
```

Figura 147. Clase; Función addData “history”.

Función que verifica la fecha actual.

```
ver(){
              if(this.verHour()){
                if(this.verDay()){
console.log( this.diasEnUnMes(this.Month,this.Year));this.Month-1;
if(this.verMonth()){this.Month=11;
this.diasEnUnMes(this.Month,this.Year) this.Year-1;} }
}else{ this.dayN();}}
```

Figura 148. Clase; Función ver “history”.

Función encargada de acceder al servicio “ServicePollutService”, guardar datos adquiridosen el array vacío “Datos”.

```
day
this.se this.CHECK()
})).subscribe(data =>let f=[];
f.push(data); for(let i of
f){
this.Datos =(i.payload.val());
}
this.onDestroy$.next();
this.onDestroy$.complete();}) }
```

Figura 149. Clase; Función dayN “history”.

Función encargada de crear la gráfica utilizando Chartjs, comprueba si existe ya una, parano solapar una sobra otra gráfica, se asignan las propiedades requeridas para pintar la gráfica y sus datos.

```
createChart(){
if(this.chart){
}
else{
this.chart = new Chart('canvas',{type:"bar",
data:{
labels:this.HoraD,datasets:[{
data:this.Values, backgroundColor:
this.Colors,
borderColor: [],borderWidth: 1
}]
},
options: {legend:{
display:false
},
scales: {
yAxes: [{
ticks: {
beginAtZero: true
                    }
                }]
            }
        }
    }) }}
```

Figura 150. Clase; Función createChar “history”.

Componente segmentos.

Plantilla HTML.

Se utiliza el componente “ion-segment-button” para crear los botones que mostraran larespectiva grafica de un determinado contaminante

```
<ion-segment-button value="P.M. 10">
<ion-label>P.M. 10</ion-label>
</ion-segment-button>
```

Figura 151. Plantilla HTML “segmentos”.

Hoja de estilos.

Estilo asignado para los botones.

```
i
--b
--color:
--background-che
--color-checked: #69bb
--indicator-color : transpareborder-
radius: 10px;
margin: 10px;ion-label{
font-family: $font;
}
}
```

Figura 152. Hoja de estilo “segmentos”.

Clase.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

City: Propiedad de tipo string, guarda la locación consultada en el modal info.

```
@Input("city") city:string;
```

Figura 153. Hoja de estilo “segmentos”.

Componente tips.

Plantilla HTML.

Plantilla HTML en base a un “ion-card” siguiendo una organización para los botones se toma “ion-grid”.

Para la toma de decisiones en cuanto a recomendaciones que se deben tomar según el índice de calidad del aire, se selecciona por medio de la directiva Switch con los rangos correspondientes.

```
        <ion-
              <h2>Ries
              <p  *ngSwitchCase=
              <p *ngSwitchCase="AQI  > 50
              <p *ngSwitchCase="AQI  > 100  && AQI <=
              <p *ngSwitchCase="AQI  > 150  && AQI <= 200">Muy
              <p *ngSwitchCase="AQI  > 201  && AQI <= 300">Extremadament
</ion-col>
```

Figura 154. Plantilla HTML; Switch “tips”.

Hoja de estilos.

Estilo asignado al componente carta de Ionic.

```
ion-card{
position: relative;background:
#0085FF;width: 90%;
left: 5%;
border-radius: 5px;padding: 0px
0px; margin: 15px 0px;
ion-card-title{ position:
relative;top: 10px;
width: 90%;
left: 5%;
text-align: center; background-color:
white;border: 2px solid black;border-
radius: 5px; color: #69BB7B;
font-size: 14px; font-family:
$font;padding: 6px 0px;}}
```

Figura 155. Hoja de estilo “tips”.

Clase.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

AQI: Propiedad de tipo cualquiera, guarda el índice de calidad del aire.

```
@Input("AQI") AQI:any;
```

Figura 156. Clase; Input “tips”.

Vista Modal text.Módulo

de rutas

Ninguna modificación en el archivo.

Plantilla HTML.

Plantilla en base de un “ion-card”, se selecciona la información correspondiente al contaminante criterio que ha sido seleccionado por el usuario a través del “icono info” Figura49.

```
<span [
<app-info-
pollutants *ngSwitchCase="
pollutants>
<app-info-
indice *ngSwitchCase="Title == 'AQI Indice de calidadindice>
</span>
```

Figura 157. Plantilla HTML; Switch “Modal-text”.

Hoja de estilos.

Estilo asignado al componente carta de Ionic.

```
ion-card{
background: rgba(255,255,255,.1);

.title{
position: relative;margin-left: 25px;
color: #69bb7b; font-family: $font;
font-size: 16px;
}
ion-card-content{ color: #69bb7b;
font-family: $font;margin: 0px 15px;
h2{
                        margin:15px 0px
}
ul{
                        li{
                          margin: 10px 0px;
}
}}}
```

Figura 158. Hoja de estilo “Modal-text”.

Clase.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

Segment: Propiedad que guarda el nombre del contaminante

```
@Input() Segment;
@Input() Title;
```

Figura 159. Clase; Inputs “Modal-text”.

Módulo.

Se realizan las importaciones necesarias para hacer funcionar el componente y hacer usode funciones externas.

```
import {
import { InfoPollutan
pollutants.component';
import { InfoIndiceComponent } from './info-ind
import {ObtInfoIndicesService } from '../Servicios/obt-infoI
indices.service';
```

Figura 160. Módulo; imports “Modal-text”.

En el módulo fueron requeridos componentes y servicios externos, los cuales sondeclarados en las propiedades correspondientes.

```
@NgMo
imports: [
CommonModule,
FormsModule,
IonicModule,
ModalTextPageRoutingModule
],
declarations: [ModalTextPage,InfoPollutantsComponent,Info
providers:[ObtInfoIndicesService]
})
```

Figura 161. Módulo; NgModule “Modal-text”.

Componente info-indice.

Plantilla HTML.

Componente Ionic, se hace una interpolación de contenido en “i.Desc” para colocar la descripción correspondiente del contaminante criterio.

```
<ion-card-content *ngFor="let i of Info">
{{i.Desc}}
</ion-card-content>
```

Figura 162. Plantilla HTML; ion-card-content “info-indice”.

Se utiliza la directiva “For” para recorrer el arreglo que guarda la información del contaminante criterio, se accede a la propiedad del arreglo con “i.propiedad”, la cuales se encuentran en “Archivos ts pollutants”.

```
<d
<


<
<p>
</div>
<div class="G
<img src="{{i
<h2>{{i.YR}}</h2
<p>  {{i.YD}}</p>
</div>
<div class="G">
<img src="{{i.OF}}" height="
<h2>{{i.OR}}</h2>
<p>  {{i.OD}}</p>
</div>
<div class="G">
<img src="{{i.RF}}" height="60px" width="6
<h2>{{i.RR}}</h2>
<p>  {{i.RD}}</p>
</div>
<div class="G">
<img src="{{i.PF}}" height="60px" width="60px" alt="">
<h2>{{i.PR}}</h2>
<p>  {{i.GD}}</p>
</div>
<div class="G">
<img src="{{i.IF}}" height="60px" width="60px" alt="">
<h2>{{i.IR}}</h2>
<p>  {{i.ID}}</p>
</div>
```

Figura 163. Plantilla HTML; contenido “info-indice”.

Hoja de estilos.

Estilo asignado al componente carta de Ionic.

```
ion-

background:
ion-card-sub position: relative
margin: 15px 5px; color: #69bb7b;
font-family: $font;
}
ion-card-content{ color: #69bb7b;
font-family: $font;margin: 0px
15px;
}
```

Figura 164. Hoja de estilo; card “info-indice”.

Clase

Arreglo vacío es necesario para guardar las propiedades del contaminante criteriorequerido.

```
Info =[];
```

Figura 165. Clase; Arreglo info “info-indice”.

Declaración de servicio “ObtInfoIndicesService”, es necesario declarar un variable del tipode servicio para acceder a toda la funcionalidad de este mismo.

```
constructor(private obtinfo: ObtInfoIndicesService) { }
```

Figura 166. Clase; constructor “info-indice”.

```
ngOnInit() { this.obtinfo.obtf().pipe(takeUntil(this.onDestroy$),finalize(()=>{
})).subscribe(data =>{ this.Info = data; })
}
```

Figura 167. Clase; ngOnInit “info-indice”.

Componente info-pollutants.

Plantilla HTML.

Plantilla HTML en base de “ion-card”, seguido de un contenedor, en él se utiliza la directiva“Switch” utilizada para la elección de información del contaminante criterio correspondiente.

```
<ion-card-con
<div *ngSwitchCase="P == 'P.M.
</div>
</ion-card-content>
```

Figura 168. Plantilla HTML; Switch “info-pollutants”.

Hoja de estilos.

Estilo asignado al componente carta de Ionic.

```
ion-card{
background: rgba(255,255,255,.1);

.title{
position: relative;margin-left:
25px; color: #69bb7b; font-
family: $font;font-size: 16px;
}
ion-card-content{ color: #69bb7b;
font-family: $font;margin: 0px
15px; h2{
                  margin:15px 0px
}
ul{
                  li{
margin: 10px 0px;
}}}}
```

Figura 169. Hoja de estilo “info-pollutants”.

Clase.

Se importan los archivos ts correspondientes de los contaminantes criterio.

```
import {
import { PM25 } from
import { O3 } from './Pollutants/O import {
SO2 } from './Pollutants/so2';import { NO2 }
from './Pollutants/NO2';import { CO } from
'./Pollutants/CO';
```

Figura 170. Clase; imports “info-pollutants”.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

P: Propiedad que guarda el nombre del contaminante criterio.

```
@Input("P") P;
```

Figura 171. Clase; Input “info-pollutants”.

Utilizando Typescript se utilizan constantes para recuperar la propiedad “ítems” de su respectivo contaminante criterio.

```
pm10 =PM
pm25 = PM25.items;o3 =
O3.items; so2 = SO2.items;
no2 = NO2.items; co
=CO.items;
```

Figura 172. Clase; Acceso a propiedades “info-pollutants”.

Archivos ts pollutants.

Archivos que contienen objetos siendo

0: Objeto que guarda el estado de la calidad del aire.

1: Objeto que guarda el nivel de riesgo del índice de calidad del aire.

2: Objeto que guarda el rango del respectivo nivel del índice de calidad del aire.

3: Objeto que guarda la ruta de la imagen representativa del rango de calidad del aire.

Componente header.

Plantilla HTML.

Plantilla HTML en base del componente Ionic “ion-header”, se aplica un estilo personalizadoen la Hoja de estilo.

```
<ion-he
<ion-toolbar >
<ion-button  slot="start"
<ion-icon name="menu-outline"  col
</ion-button> <ion-title mode="md" color="acept
</ion-toolbar >
</ion-header>
```

Figura 173. Plantilla HTML “header”.

Hoja de estilos.

Personalización del componente utilizando su respectiva clase de estilo, reasignando elcolor de fondo “—background”.

```
ion-toolbar.md{
--background: rgba(255,255, 255,.3);


border-radius: 30px;width: 90%;
left: 5%;
}
.header{ top:10px;
}
ion-icon{
font-size:   22px;; font-weight:
800;color:green;
}
ion-title{
font-size: 18px; font-weight:
600; font-family: $font;
}
```

Figura 174. Hoja de estilo “header”.

Clase.

Declaración de “MenuController” necesario para tener el control del menú.

```
constructor(private menu: MenuController) {
```

Figura 175. Clase; constructor “header”.

Función encargada de abrir el componente menú utilizando “menú” constante declarada detipo “MenuController”

```
Open(){ this.menu.open('main');
}
```

Figura 176. Clase; Función Open “header”.

Componente menu.

Plantilla HTML.

Plantilla HTML en base del componente Ionic “ion-menu”, se aplica un estilo personalizadoen la Hoja de estilo.

```
<io
<ion-
<ion-lis
<ion-list-
<ion-note>In
<ion-menu-toggle a hide="false"
*ngFor="let p of
<ion-
item (click)="selectedIndex = i" rout
]" lines="none" detail="false" [class.sel
<ion-icon slot="start" [ios]="p.iconoutline'"
[md]="p.icon + '-sharp'"></ion-icon>
<ion-label>{{ p.title }}</ion-label>
</ion-item>
</ion-menu-toggle>
</ion-list>
</ion-content>
</ion-menu>
```

Figura 177. Plantilla HTML “menu”.

Hoja de estilos.

Personalización del componente utilizando su respectiva clase de estilo, se reasignanvarios valores.

```
ion-
--bac
}
ion-menu ion-conte
--background: rgba(varborder-
radius: 40px ;
}

     ion-menu.md ion-content {
       --padding-start: 8px;
--padding-end: 8px;
--padding-top: 20px;
--padding-bottom: 20px;
}
```

Figura 178. Hoja de estilo “menu”.

Clase.

Arreglo encargado de guardar las propiedades de cada ruta de páginas de la aplicación.

```
public appPages = [
{
title: 'Sitios',url:
'/sitios', icon: 'globe'
},

];
```

Figura 179. Clase; Array appPages “menu”.

Componente Shared indice.

Plantilla HTML.

Plantilla HTML en base al componente Ionic “ion-grid”, se aplica personalización en la Hojade estilo.

```
<


circle-

<im
</ion-
</ion-co
<ion-col
<ion-row>
                 <ion-col s
                   <ion-item
                 <ion-label >
{{AQI}}
</ion-label>
</ion-item>
</ion-col>
</ion-row>
<ion-row>
<ion-col size="12">
<ion-item lines="none" >
<ion-label  class="ion-text-wra
</ion-item>
</ion-col>
</ion-row>
</ion-col >
</ion-row>
<ion-row  class="White">
<ion-col size="4">
<ion-icon  name="sunny"  ></ion-icon>
<ion-label  >{{Temp}}°</ion-label>
<hr width="1"  size="40">
</ion-col>
<ion-col size="4" >
<ion-icon  name="water-outline"  ></ion-icon>
<ion-label  >{{Humedad}}%</ion-label>
<hr width="1"  size="40">
</ion-col>
<ion-col size="4" >
<ion-icon name="send-outline"  ></ion-icon>
<ion-label  >{{Viento}}  km/h</ion-label>
<hr width="1"  size="40">
</ion-col>
</ion-row>  </ion-grid>
```

Figura 180. Plantilla HTML “shared”.

Hoja de estilos.

Personalización respectiva con la condición del nivel de calidad del aire.

```
.R
po backg
border-r
}
.Row-I-Aceptable{ position:
relativ background: rgba(255
border-radius: 10px;
}

.Row-I-Mala{
position:  relative; background:
rgba(255,126,0,.9);border-radius: 10px;
}
.Row-I-Muy-Mala{ position:
relative;
background: rgba(255,0,0,.9);border-
radius: 10px;
}
.Row-I-Ex-Mala{ position:
relative;
background: rgba(143,63,151,.9);
border-radius: 10px;
}
```

Figura 181. Plantilla HTML “shared”.

Clase.

Los @Input son utilizados para declarar las propiedades del componente, estos mismosson utilizados para realizar una interpolación, pasarlos a un subcomponente, servicio o pipe.

AQI: Propiedad de tipo cualquiera, guarda el índice de calidad del aire.

Class-c: Propiedad de tipo cualquiera, guarda la clase del estilo respectivo a la condiciónde la calidad del aire.

Face-c: Propiedad de tipo cualquiera, guarda la selección de la imagen respectiva a la condición de la calidad del aire.

Humedad, Viento, Temp: Propiedades de tipo cualquiera, guarda las variablesmeteorológicas para mostrar en su respectiva localidad y fecha.

```
@Input( @Input("Class-
c")
@Input("Face-c") Face:string;
@Input("Calidad-c") Calidad:string;
@Input("Humedad") Humedad:string;
@Input("Viento") Viento:string;
@Input("Temp") Temp:string;
```

Figura 182. Clase; Inputs “shared”.

Componente app.Módulo de Rutas

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
```

Figura 183. Módulo de rutas; import “app”.

Se importan los módulos necesarios para hacer funcional las rutas que se declaran.

```
var iniciovar r;
(()=>{
r = localStorage.getItem('2'
if(r==null){
inicio = "slides";
localStorage.setItem("2","true");
}else{
inicio = "sitios";

}
})();
```

Figura 184. Módulo de rutas; Función anónima “app”.

Var inicio: Variable de la ruta en la que inicia la aplicación.Var r:

Variable que obtiene el item “2” del localstorage.

Se ejecuta una función anónima automáticamente, dentro de la función se encuentra un condicional if/else

Su lógica es verificar si el item “2” de localstorage es null o contiene algún dato.

En caso de ser verdadero se asigna como vista principal “slides” y asigna al item “2” un“true”

Caso contrario se asigna como vista principal “sitios”.

```
const routes: Routes = [
{
path: '', redirectTo: inicio,
pathMatch: 'full'
},
{
path: 'sitios',
loadChildren: () => import('./sitios/sitios.module').then( m => m.Siti
osPageModule)
},
{
path: 'slides',
loadChildren: () => import('./slides/slides.module').then( m => m.Slides
PageModule)
},
{
path: 'modal-text',
loadChildren: () => import('./modal-text/modal-
text.module').then( m => m.ModalTextPageModule)
}];
```

Figura 185. Módulo de rutas; Routes “app”.

La declaración de la constante routes es de tipo Routes, se encuentran las rutas disponiblespara navegar por la aplicación.

Plantilla HTML.

Plantilla HTML principal, compuesta por “ion-app”, el cual se declara como tal, se colocanlos componentes a mostrar en vista principal.

```
<ion-app>
<app-menu></app-m
<ion-content >
<ion-router-outlet id="main-content"></ion-
</ion-content>
</ion-app>
```

Figura 186. Plantilla HTML “app”.

Hoja de Estilos.

No se declaró ninguna clase para este componente.

Clase.

```
import { Component, OnInit } from '@angular/core';import {
Platform } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';import {
StatusBar } from '@ionic-native/status-bar/ngx'; @Component({
selector:  'app-root', templateUrl:
'app.component.html',styleUrls:
['app.component.scss']
})
export class AppComponent implements OnInit {
constructor(
private platform: Platform,
private splashScreen: SplashScreen,private
statusBar: StatusBar,
) {
this.initializeApp();
}

initializeApp() {
this.platform.ready().then(() => {
this.statusBar.styleDefault();
this.splashScreen.hide();

});
}
ngOnInit() { }
}
```

Figura 187. Clase “app”.

La función que cumple la clase de app es lanzar el “statusBar” por defecto y “splashScreen” generado dinámicamente al comienzo del proyecto.

Módulo

```
const firebaseConfig = {
apiKey:  "AIzaSyDb7oZFCLb0ZJ6l6WkHcl_VECoXHxpSEcI",
authDomain: "airnl-58f75.firebaseapp.com", databaseURL:
"https://airnl-58f75.firebaseio.com",projectId: "airnl-
58f75",
storageBucket: "airnl-58f75.appspot.com",
messagingSenderId: "200150779174",
appId: "1:200150779174:web:47520130de4bca23632de7",
measurementId: "G-TYB4YFRS68"
};
```

Figura 188. Módulo; Firebase Configuración “app”.

Se declara una constante con las propiedades necesarias para configurar la conexión con Firebase.

```
@Ng
decl entryCo
imports: [
BrowserModu IonicModule.fo
IonicStorageModule driverOrder:
['indexeAppRoutingModule,
FormsModule, HttpClientModule,
AngularFireModule.initializeApp(fiAngularFireDatabaseModule

],
providers: [ StatusBar,
SplashScreen,HTTP,
{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy

],
bootstrap: [AppComponent]
})
```

Figura 189. Módulo; NgModule “app”.

Estas son las propiedades que se declaran en el módulo.

declarations: Se declaran los componentes que se utilizaran en la plantilla HTML.

imports: Se importan los módulos necesarios para el funcionamiento de la aplicación, jerárquicamente “app.module.ts” se encuentra encima de todos los módulos por lo tanto, comparte estos módulos con jerarquías bajas.

providers: Se declaran los servicios y pantallas de carga al iniciar la aplicación.

Servicios

Los servicios generalmente son utilizados para escribir lógica de esta manera mantenemosnuestros componentes y page limpias, sin embargo, es de suma importancia tener un servicio para cada tipo de función, en mi proyecto asigne un servicio a cada tipo de funcióndiferente, todos realizan peticiones utilizando el protocolo http.

Una buena práctica al realizar una petición de datos es la creación de interface, que nos permiten tener un molde de los datos de nuestro BD, ya sea unitarios o un grupo de datos,convirtiendo nuestra interface en un arreglo.

Función que retorna un arreglo de tipo observable, su lógica es realizar un http request en localhost del dispositivo para obtener el archivo json requerido en su respectiva ruta.

Todos los servicios están conformados por el siguiente formato.

Obt-estados

Obtención de estados disponibles.

```
obtE(Pais):
return this.http.get<Estados[]>(");
}
```

Figura 190. Servicio; Obt-estados “Servicio”.

Obt-infoindices

Obtención de información de los índices de calidad del aire.

```
obtf(): Observable<Indice[]>{
return this.http.get<Indice[]>("../assets/Infos/Info-Indices.json");
}
```

Figura 191. Servicio; Obt-infoindices “Servicio”.

Obt-municipios

Obtención de municipios disponibles.

```
obtm(P,Est):Observable<Municipios[]>{
return this.http.get<Municipios[]>("../assets/Localidades/"+P+"/"+Est+"/"+Est+".json");
}
```

Figura 192. Servicio; Obt-municipios “Servicio”.

Obt-paises

Obtención de países disponibles.

```
obtpais():O
return  this.http
.get<Pais[]>("../../assets/Localidades/Pais.jso
}
```

Figura 193. Servicio; Obt-paises “Servicio”.

Obt-pollutants

Obtención de la base de datos en Firebase retorna una promesa, función clave para obtenerla información necesaria de los contaminantes criterio y variables meteorológicas.

```
obtfire()
this.itemref = this.db.obj
let d= this.itemref.snapshotChanges();return d;
}
```

Figura 194. Servicio; Obt-pollutants “Servicio”.

Pipes

Los pipes son utilizados para realizar filtros de datos, en la aplicación son muy usados especialmente para filtrar la elección del color al que se asocia el rango según el contaminante criterio.

Pipes contaminantes criterio.

Siguiendo NORMA Oficial Mexicana NOM-172-SEMARNAT-2019 para la asignación de colores en cuestión con los rangos del índice de la calidad de aire respectivamente de cadacontaminante criterio, se aplica la función “transform” encargada de realizar el filtro se sigueun formato de recorrer el arreglo de datos de cada contaminante, por medio de un “Switch”se hace la selección del color.

El formato para la asignación de color es RGB(red,green,blue).

CO.

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
transform(value: any) {let color =
[];
for( let items of value){switch
(true) {
case items <= 8.75: color.push("rgb(0,228,0)");
break;
case items > 8.75 && items <=11.00:
color.push("rgb(255,255,0)"); break;
case items >11.00 && items <=13.30:
color.push("rgb(255,126,0)"); break;
case items >13.30 && items <=15.50:
color.push("rgb(255,0,0)"); break;
case items >15.50: color.push("rgb(143,63,151)");
break;
default:break;
}
}
return color;
}
```

Figura 195. Pipes; CO “Pipes”.

NO2.

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
transform(value: any) {let
color = [];
for( let items of value){switch
(true) {
case items <= .107:
color.push("rgb(0,228,0)");break;
case items >.107 && items <=.210:
color.push("rgb(255,255,0)"); break;
case items >.210 && items <=.230:
color.push("rgb(255,126,0)"); break;
case items >.230 && items <=.250:
color.push("rgb(255,0,0)"); break;
case items >.250: color.push("rgb(143,63,151)");
break; default:
break;
}
}
return color;
}
```

Figura 196. Pipes; NO2 “Pipes”.

O3.

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
transform(value: any) {let
color = [];
for( let items of value){switch
(true) {
case items <= .051: color.push("rgb(0,228,0)");
break;
case items >.051 && items <=.095:
color.push("rgb(255,255,0)"); break;
case items >.095 && items <=.135:
color.push("rgb(255,126,0)"); break;
case items >.135 && items <=.175:
color.push("rgb(255,0,0)"); break;
case items >.175: color.push("rgb(143,63,151)");
break; default:
break;
}
}
return color;
}
```

Figura 197. Pipes; O3 “Pipes”.

P.M. 10

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
transform(value: any) {let
color = [];
for( let items of value){switch
(true) {
case items <= 50:{ color.push("rgb(0,228,0)");
break;
}
case items >50 && items <=75:{ color.push("rgb(255,255,0)");
break;
}
case items >75 && items <=155:{
color.push("rgb(255,126,0)");
break;
}
case items >155 && items <=235:{
color.push("rgb(255,0,0)"); break;
                                                    }
            case items >=236:{
color.push("rgb(143,63,151)");break;
}
default:break;
}
}
```

Figura 198. Pipes; P.M. 10 “Pipes”.

P.M. 2.5

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
t
le for(
switch
case it color.pubreak;
case items >25 &&
color.push("rgb(25break;
case items >45 && items <=
color.push("rgb(255,126,0)")break;
case items >79 && items <=147:
color.push("rgb(255,0,0)"); break;
case items >=147: color.push("rgb(143,63,151)");
break;
default:break;
}

}
```

Figura 199. Pipes; P.M. 2.5 “Pipes”.

SO2.

Pipe encargado de clasificar los datos obtenidos de la base de datos, se clasifican los datosdesde la hora actual hasta la hora 00:00 del día.

```
t
le for(
switch
case ite color.pusbreak;
case items >.008 &
color.push("rgb(255break;
case items >.110 && items <=
color.push("rgb(255,126,0)");break;
case items >.165 && items <=.220:
color.push("rgb(255,0,0)"); break;
case items >.220: color.push("rgb(143,63,151)");
break;
default:break;
}
}
```

Figura 200. Pipes; SO3 “Pipes”.

Pipe USAQI

Pipe encargado de realizar el cálculo del índice de calidad del aire siguiendo la fórmula deAQI Figura 34 del documento EPA 454/B-18-007.

Se utiliza la función “transform” para realizar el cálculo, se utiliza un “switch” para elegir elrango del nivel de la calidad del aire y realizar correctamente el cálculo.

El resultado es redondeado al valor más cercano, siendo que es mayor a .5 se redondea alnúmero siguiente.

```
transform(value:any,hora:any) {let PM2 ;
if(hora.split(":",1)<10){
hora =0+""+hora;
}else{}
for(let element of value){ for(let items of
element){if(items.Hora == hora){
PM2 = items.PM25;
}
}
}
let k;let l;
let indice;
switch (true) { case PM2 <= 12.0:
k = (50-0)/(12.0-0);
indice = k *(PM2-0)+0;break;
case PM2 >=12.1 && PM2 <= 45.4 :
k = (100-51)/(35.4-12.1);
indice = k *(PM2-
12.1)+51; // k *(indice contaminante - indice ICA) + RANGO MENOR INDICEbreak;
case PM2 >= 35.5 && PM2 <= 55.4:k = (150-101)/(55.4-35.5);
indice = (k*(PM2-35.5))+101;break;
case PM2 >= 55.5 && PM2 <= 150.4:k = (200-151)/(150.4-
55.5);
indice = k *(PM2-55.5)+151;break;
case PM2 >= 150.5 && PM2 <= 250.4:k = (300-201)/(250.4-
150.5);
indice = k *(PM2-150.5)+201;break;
case PM2 >= 250 && PM2 <=500.4:k = (500-301)/(500.4-
250.5);
indice = k *(PM2-250.5)+301;break;
default:break;
}
return Math.round(indice);
```

Figura 201. Pipes; USAQI “Pipes”.

Pipe city.

Pipe encargado de evaluar los datos recibidos de Firebase y obtener únicamente el valorde la hora y fecha actual.

```
tr
const if(arg.split( arg
=0+""+arg;

}else{}
for(const post of value){
if(post.Hora == arg){
res.push(post);
}
}
return res;
}
```

Figura 202. Pipes; city “Pipes”.

Pipe graphic.

Pipe encargado de evaluar los datos recibidos de Firebase y obtener únicamente los valoresde la actual hasta la hora 00:00.

```
transform(value:any, arg: any,S:any) {let res=
[];
this.resv =[]; for(let i of
value){
if(i.Hora.split(":",1) <= arg){

res.push(i.Hora);

this.resv.push(i[S]);
}
}
return res;
}
```

Figura 203. Pipes; graphic “Pipes”.

Pipes índices.

Cada contaminante criterio cuenta con su pipe para evaluar el índice, de esta forma seobtiene la condición del índice.

Siguiendo NORMA Oficial Mexicana NOM-172-SEMARNAT-2019.

Se muestra el formato a seguir para cada uno de los contaminantes criterio.

Se selecciona el color respectivamente con el rango del índice de calidad del aire del contaminante.

Se retorna el color.

```
t
le
swi
case
color=break;
case value > 8. color="Yellow";
break;
case value >11.00 && valu
color="Orange";
break;
case value >13.30 && value <=15.50:color="Red";
break;
case value >15.50:
color="Purple"; break;
default:break;
}
return color;
}
```

Figura 204. Pipes; indices “Pipes”.

Metodología de análisis de herramientas.

En la planeación del ciclo de vida del proyecto se determina el tipo de software a desarrollarde acuerdo al uso del mismo; en nuestro caso se trata del desarrollo de una aplicación móvil; en base a esto se realizó un análisis de herramientas con diferentes tecnologías disponibles para el ámbito de aplicación.

Se realizaron tablas comparativas con evaluación de criterios de calidad del sofware seleccionados para cada herramienta de desarrollo. Una vez hecho el análisis de las herramientas disponibles, se determinan cuáles son adecuadas para el proyecto.

Se utiliza el modelo presentado por Hewlett en 1987, el cual es conocido con el acrónimo "FURPS" y define los factores de calidad para el desarrollo de software.

El acrónimo está compuesto de las siguientes palabras:

Funcionalidad (Functionality).

Usabilidad (Usability). Confiabilidad

(Reliability). Presentación (Perfomance).

Soporte (Supportability).

La descripción de estos conceptos se puede ver en la siguiente tabla:

Tabla 06. Modelo Furps

Sigla	Tipo de Requerimiento		Descripción
F	Funtional	***Funcional***	Características, capacidades y algunos aspectos de seguridad
U	Usability	***Facilidad de Uso***	Factores Humanos (interacción), ayuda, documentación
R	Reliability	***Fiabilidad***	Frecuencia de fallos, capacidad de recuperación de un fallo y grado de previsión
P	Performance	***Rendimiento***	Tiempos de respuesta, productividad, precisión, disponibilidad, uso de los recursos
S	Supportability	***Soporte***	Adaptabilidad, facilidad de mantenimiento, internacionalización, facilidad de configuración.

CAPITULO IV. RESULTADOS.

4.1 Análisis

Se realiza el análisis de herramientas según su enfoque en el desarrollo del proyecto de software.

Existen dos enfoques dentro del desarrollo de aplicaciones móviles hibridas: el Front-end yel Back-end. Ambos enfoques se analizan por separado según las herramientas disponibles para cada enfoque y se clasifican en función del binomio costo-beneficio.

El Front-end es el desarrollo realizado del lado del cliente. El Back-

end es el desarrollo realizado del lado del servidor.

Análisis de herramientas Front-end.

Tabla 07. Análisis de herramientas Front-end.

Indicadores FURPS					
Framework	**Funciona-lidad**	**Facilidad de uso**	**Fiabilidad**	**Rendimiento**	**Soporte**
Angular JS (JavaScript)	Permite la utilización de información dinámica, para aplicaciones tanto web y móviles hibridas. Utiliza JQuery Lite que permite la manipulación de elementos DOM (Document Object Model), sin la necesidadde añadir bibliotecas externas, disponibilida d de herramientas como Pipes se utilizan tanto para realizar funciones de filtrado y conversión,	Es difícil cuando un programador novato la utiliza, este debe de aprender Typescript para suuso. Los modelos usados son objetos planos de JavaScript, da mayor facilidad enla transferencia de datos entre aplicaciones y servicios REST (Representational State Transfer).	Cuenta con una herramienta de integración de independencia s al momento de realizar actualizaciones del Framework.	Se renderiza el mismo código de manera distinta tanto en web como aplicación móvil hibrida. Tiempos de carga rápidos y eficiente en carga de bibliotecas.	Las guías de instalación y de inicio son fáciles de comprender. Cuenta con herramientas que ofrecen facilidad de sincronizar modelos con HTML5. Actualizaciones continuas y recomendacio nes para mejorar proyectos.

	con los Servicios se realizan peticiones a servidores o escribir la lógica de un componente.				
React Native	Rendimiento mejorado enel renderizado de pantallas, brinda la posibilidad de pasar funciones como parámetros entre servicios, incluso es posible crear componentes en base a funciones. Depende de librerías externas para realizar funciones complejas, como filtración de datos y peticiones al servidor.	Se cuenta con guías para la reducción de la curva de aprendizaje. Escalabilidad simple, facilidad de mantener y depurar.	Utiliza JavaScript para escribir código. Ofrece actualizaciones DOM (Document Object Model), con cada cambio de realizado en un determinado componente de la vista. Se virtualiza el DOM como ventaja.	El código ejecutado se realiza mediante el DOM (Document Object Model) virtual, generando mayor eficienciay rapidez.	La facilidad que tiene con los componentes individuales es en cuanto la migración y mantenimientoa otro Framework, implicaría reescribir la mayor parte del código HTML.
Ionic	Facilita la creación de aplicaciones móviles hibridas con tecnología web. Ofrece plantillas para las plataformas Android e iOS	Se cuenta con guías muy sencillas en su sitio web oficial, las plantillas estánbasadas en etiquetas HTML la implementación y configuración es sencilla de realizar.	Las plantillas para implementar están testeadas en lamayoría de teléfonos móviles.	Las plantillas son totalmente responsivas lo que significa quese adaptan al tamaño de la pantalla del dispositivo. Son ligeras al momento de renderizar una pantalla.	Se realiza una continua actualización del Framework además cuenta con una herramienta disponible para realizar la actualización sin perder las dependencias utilizadas.

Análisis de herramientas Back-end.

Tabla 08. Análisis de herramientas Back-end.

Indicadores FURPS					
Framework	**Funcionalidad**	**Facilidad de uso**	**Fiabilidad**	**Rendimiento**	**Soporte**
Firebase	Alojamiento de base de datos no SQL en la nube en tiempo real, almacenados en formato JSON.	Documentación disponible para diferentes ámbitos, es decir, para web y Frameworks.	Al momento de crear la base de datos en cuenta propia, se asignan reglas de quien puede acceder, modificar y obtener registros de esta misma.	Utiliza SDK de Performance Monitoring para recopilar datos de rendimiento y realizar mejoras en el rendimiento de la aplicación.	Rápida configuración, con un objeto creado en el código está listo para acceder a los métodos disponibles. Con el mismo objeto es posible cambiar la configuración de la base de datos objetivo.
MySQL	Capacidad de realizar múltiples consultas, soporta hasta 32 índices de tablas diferentes. Ingreso en masa de datos por columna.	Se cuenta con guías para la reducción de la curva de aprendizaje. Escalabilidad simple, facilidad de mantener y depurar.	Características de seguridad, como la autenticación de base de datos y a su ingreso, proporciona soporte SSH y SSL, para conexiones más seguras.	Arquitectura de almacenamiento que permite a los Administradores de base de datos realizar una configuración sencilla, alta velocidad en peticiones, con la capacidad de distintas caches de memoria.	Maneja aplicaciones arraigadas con huellas de al menos 1 MB a la ejecución de almacenamientos de datos, se maneja con varias versiones de Linux, Unix y Windows.
MongoDB	Las consultas son realizadas mediante objetos JSON como parámetros, la información es almacenada en BSON y es base de datos NO SQL.	Documentación fácil de entender con una curva de aprendizaje sencilla.	La integración de código es muy fiable sin embargo se debe de tener cuidado al momento de definir la base de datos y sus campos no relacionados, puede surgir un error al momento de escribir los campos.	La escala de información pobre al momento de llegar a cierto volumen se reduce el rendimiento del procesamiento de información.	La migración de versiones es sencilla y cuenta con un soporte muy bueno al momento de realizar migraciones de base de datos y cambio de versión y formato.

Discusión del análisis.

Se puede observar en la Tabla 07, que Angular es muy completo y no depende de librerías externas para realizar funciones complejas como filtrado y conversión de **datos,** Angular ofrece "Pipes" para realizar estas tareas, y además ofrece una herramienta denominada "Services", que permite añadir los servicios que sean necesarios, en ellos seescribe la lógica y peticiones a servidores.

React Native ofrece mayor eficiencia y rendimiento, pero depende de librerías externas para realizar funciones complejas como filtrado y conversión de datos, y realizar peticionesa servidores.

Ionic es seleccionado porque ofrece plantillas de componentes estándar de Android & iOSy es compatible con el desarrollo de aplicaciones móviles hibridas.

Se eligen Angular e Ionic para el desarrollo de la aplicación móvil hibrida, porque en conjunto son las mejores herramientas; Ionic es utilizado como promotor de plantillas de componentes para Angular con el motivo de cumplir con los diseños estándar para Android& iOS, por otra parte, Angular es seleccionado por las cuestiones de lógica y creación de pantallas.

Como lo muestra la Tabla 08**, Firebase** se utiliza sobre las demás herramientas por su accesibilidad y soporte a la comunidad, además de que brinda su base de datos en tiemporeal y en formato JSON.

Figma es una herramienta gratuita para el diseño de prototipos de interfaces de usuario, esmuy sencilla de utilizar, ofrece una base de datos de imágenes, edición de imágenes, colaboración de equipo en tiempo real.

JSON es el formato utilizado para el intercambio de datos con Firebase, brinda facilidad delectura y escritura de información.

JavaScript lenguaje de programación utilizado para escribir código en Angular, es utilizado en archivos con extensión .js que contiene JavaScript Vanilla, es decir, sin la combinación de TypeScript.

TypeScript lenguaje de programación superconjuntado de JavaScript, utilizado para escribir código en Angular en conjunto con la funcionalidad de JavaScript, nos permite escribir código de tipos estáticos y objetos basados en clases.

Herramientas de Framework para desarrollar la App:

- Angular e Ionic

Herramientas para diseño de interfaces de usuario:

- Figma

Herramientas para base de datos:

- FireBase

Herramientas para el modelo de datos:

- JSON

Herramientas para el código fuente:

- JavaScript
- TypeScript

Capitulo V. CONCLUSIONES Y TRABAJOS FUTUROS.

Conclusiones

En base a la aplicación móvil desarrollada para el monitoreo de contaminantes criterio, se tuvo la experiencia de realizar un comparativo de análisis de herramientas para el desarrollode aplicaciones móviles tanto hibridas como nativas, de este modo se toma la decisión óptima en cuanto al tipo de herramientas a utilizar para el desarrollo del proyecto.

El objetivo de la tesis: "*Analizar herramientas actuales para el desarrollo de aplicaciones móviles hibridas aplicando un desarrollo de software en un sistema de tiempo real para el seguimiento de contaminantes criterio*". Dicho objetivo se logró analizando las herramientas mediante su documentación técnica y comparándola con herramientas de desarrollo de aplicaciones nativas. Además, se hizo uso de las herramientas seleccionadas en un proceso desarrollo de una app para el monitoreo de contaminantes en el área metropolitana de Monterrey, generando mayor conocimiento en la práctica.

El objetivo es logrado realizando el análisis a nivel técnico, y seleccionando las herramientas con mejor funcionalidad, facilidad de uso, fiabilidad y soporte, siendo elegidos Angular e Ionic del desarrollo de aplicaciones móviles hibridas.

Se concluye que Angular es la mejor herramienta para el desarrollo de la aplicación utilizando sus principales características para realizar la lógica, conexión de componentes y contenedores de vistas del usuario, al usarlo junto a Ionic se complementa el proyecto yaque ofrece un estándar de diseño para los componentes tanto para Android como para iOS.

La hipótesis de la presente investigación, "Los frameworks para aplicaciones móviles hibridas cumplen con la misma calidad para el proceso de desarrollo de software con respecto a frameworks de desarrollo móvil nativo".

Aun cuando las aplicaciones desarrolladas de manera hibrida se ejecutan mediante un webview, éstas tienen el mismo rendimiento que una aplicación nativa que no requiere de un web view.

Por definición al utilizar una web view dentro de una aplicación móvil su rendimiento hace que sea menor, sin embargo, en la actualidad este problema es totalmente minimizado utilizando UI (User Interface) formada por controles nativos configurados y personalizados con CSS Y un SGML (Standard Generalized Markup Language) el estándar para definir lenguajes de marcado generalizados, utilizando HTML como lenguaje de marcado.

Se trabajó con Angular un framework de desarrollo híbrido el cual ofrece un estándar de estructura de proyecto aplicando MVC (Modelo, Vista y Controlador).

Se trabajó además con Ionic, el cual ofrece diseños de componentes estándar para las plataformas de Android e iOS, el estándar de componentes está basado en las normativascorrespondientes a cada plataforma.

En Android es utilizado "Material Design" enfoque de diseño creado por Google, para iOS se sigue "Human Interface Guidelines" documentación ofrecida para los desarrolladores y diseñadores de aplicaciones en conjunto, lo que agrega calidad de diseño al proyecto. En contraste, el framework para desarrollo nativo React Native, no ofrece un estándar de estructura definido de proyecto, lo cual para programadores inexpertos resulta un problema

al momento de definir la estructura del proyecto, este problema afecta al momento de buscar la locación de un componente o servicio, en caso de no realizar correctamente la definición de estructura del proyecto.

Para obtener un diseño de componentes estándar para las plataformas móviles es necesario realizar importaciones de librerías externas, lo que puede resultar un problema en la compatibilidad con otras librerías, en caso de realizar los diseños desde cero se consumiría mucho tiempo para su desarrollo.

Se realizaron pruebas del software en teléfonos móviles teniendo como resultado:

1) El tiempo de carga es casi nulo al momento de mostrar las pantallas programadas.
2) En el proceso de conversión del código al formato apk, se realiza la compilación del código, la compresión del mismo y su optimización.
3) La curva de aprendizaje en el ámbito teórico resulta difícil de comprender sin tener una noción del funcionamiento y conceptos básicos, sin embargo, al momento de adentrarse en la práctica con ejercicios simples, tales como el envío de información a través de un formulario, obtención de información de una base de datos y plasmar la información en una tabla, complementa el entendimiento a conceptos de los frameworks Angular e Ionic. Los conceptos se refieren a la metodología que es utilizada para hacer la creación de la aplicación, tales como palabras reservadas para realizar ya sea una compilación de componentes o complementación de algún componente y vista.
4) El desarrollo de interfaces para usuario resulta sencillo debido a la integración de Ionic que nos ofrece un diseño estándar de componentes para cada plataforma móvil, los componentes son tales como un menú, barra de títulos, modales, entre otros componentes clásicos.

Trabajos Futuros.

Como cualquier otro trabajo de tesis y proyecto de investigación pueden existir diversos caminos abiertos los cuales es posible investigar aún más y seguir trabajando en ellos.

Durante este trabajo de tesis me han surgido distintas ideas y caminos a seguir para futuras investigaciones, la mayoría son relacionadas directamente con este trabajo, y otras son el resultado de dudas y cuestiones, algunas otras no se relacionan con la tesis, sin embargo, pueden ser tomadas para posteriores investigaciones.

Un trabajo de investigación futura sería el de comparar los frameworks híbiridos y nativos en relación a la usabilidad, el tiempo de desarrollo de software y uso de recursos computacionales para aplicaciones móviles.

Un trabajo de investigación más, sería respecto de la aceptación y adopción de estos frameworks en el medio profesional de producción de software.

El uso de frameworks híbridos puede aplicarse a:

- Análisis del desarrollo web de la usabilidad de frameworks y lenguajes de programación nativos en el contexto de un sistema de seguimiento de tickets para soporte a clientes.
- Análisis de herramientas para el desarrollo de una aplicación móvil nativa en el contexto del e-commerce.
- Análisis de herramientas Back-end para el desarrollo de una API pública en el contexto de seguimiento de compra y venta en general.

REFERENCIAS Y FUENTES DE INFORMACIÓN

Sube una app - Ayuda de Play Console. (s. f.). Support Google. Recuperado 6 de octubrede 2020, de https://support.google.com/googleplay/android- developer/answer/113469?hl=es-419&ref_topic=7072031

Ionic - Cross-Platform Mobile App Development. (s. f.). Ionic Framework. Recuperado 6 deoctubre de 2020, de https://ionicframework.com/

Ionic Framework - Ionic Documentation. (s. f.). Ionic Docs. Recuperado 6 de octubre de2020, de https://ionicframework.com/docs

UI Components - Ionic Documentation. (s. f.). Ionic Docs. Recuperado 6 de octubre de2020, de https://ionicframework.com/docs/components

Angular Documentation. (s. f.). Angular Docs. Recuperado 6 de octubre de 2020, de https://angular.io/docs

Design, prototype, and gather feedback all in one place with. (s. f.). Figma. Recuperado 6de octubre de 2020, de https://www.figma.com/design/

Free Prototyping Tool to Create Clickable Prototypes. (s. f.). Figma. Recuperado 6 deoctubre de 2020, de https://www.figma.com/prototyping/

El Financiero. (2018, 2 febrero). *Monterrey, la ciudad más contaminada*. https://www.elfinanciero.com.mx/monterrey/la-ciudad-mas-contaminada

Trabajando con JSON. (s. f.). Documentación web de MDN. Recuperado 1 de octubre de2020, de https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON

Communicating with backend services using HTTP. (s. f.). Angular HTTP. Recuperado 28de septiembre de 2020, de https://angular.io/guide/http

Publicado por: Andrea Cantú. (s. f.). *Qué es: UX y UI*. Andrea Cantú. Recuperado 6 deoctubre de 2020, de https://blog.acantu.com/que-es-ux-y-ui/

Cómo definir el ciclo de vida del desarrollo de software móvil. (s. f.). Yeeply. Recuperado 7de octubre de 2020, de https://www.yeeply.com/blog/ciclo-de-vida-desarrollo-software- movil/

Fluid UI Features. (s. f.). Fluid UI. Recuperado 7 de octubre de 2020, de https://www.fluidui.com/features

Siripathi, S. (2017, 26 octubre). Herramientas de Desarrollo Móvil. Code Envato Tuts+. https://code.tutsplus.com/es/articles/mobile-development-tools--cms-29792

Cómo comenzar a usar el NDK | NDK de Android |. (s. f.). Android Developers. Recuperado 8 de octubre de 2020, de https://developer.android.com/ndk/guides?hl=es-419

Apple Developer Documentation. (s. f.). Apple Developer. Recuperado 8 de octubre de2020, de https://developer.apple.com/documentation/

Download Android Studio and SDK tools |. (s. f.). Android Developers. Recuperado 8 deoctubre de 2020, de https://developer.android.com/studio

Acerca. (s. f.). Node.js. Recuperado 8 de octubre de 2020, de https://nodejs.org/es/about/

SIMA Nuevo León (2017). AireNL (Versión 1.0)[Aplicación Móvil].Descargado de: https://play.google.com/store/apps/details?id=com.icalialabs.airenl&hl=nl

IQAir AG (2020). Calidad del Aire | AirVisual (Versión 5.50.0-14.3) [Aplicación Móvil] Descargada de: https://play.google.com/store/apps/details?id=com.airvisual&hl=es_MX

Dirección de Monitoreo Atmosférico (2020). Aire (Versión 2.3.30) [Aplicación Móvil] Descargada de: https://play.google.com/store/apps/details?id=mx.gob.df.aire&hl=es_MX

SIMA Nuevo León (2020). Aire y salud NL (Versión 1.0) [Aplicación Móvil] Descargado de: https://play.google.com/store/apps/details?id=mx.gob.nl.AireySaludNL

Office of Air Quality Planning and Standards. (2018, septiembre). *Technical Assistance Document for the Reporting of Daily Air Quality – the Air Quality Index (AQI)* (N.º 1). Nan.https://www.airnow.gov/sites/default/files/2020-05/aqi-technical-assistance-document- sept2018.pdf