

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DISERTACION PREVIA A LA OPTENCION DEL TITULO DE
INGENIERO DE SISTEMAS

“DESARROLLO DE UN APLICATIVO MÓVIL PARA CÁLCULOS
BÁSICOS PETROLEROS REQUERIDO POR LA EMPRESA SERTECPET
S.A”

NOMBRE(S)

JEAN ANDRE GONZALEZ CERON

DIRECTOR/A: ANDRES JIMENEZ

CIUDAD: QUITO, AÑO: 2016

1. Introducción

1.1. Dedicatoria

Dedico este trabajo a mi familia quienes estuvieron brindándome su incondicional apoyo durante estos años de estudio, a mis padres por su constante guía, a mi esposa y a mis hijos por ser el impulso para afrontar contratiempos y altercados, a mis compañeros por acogerme como un hermano durante este tiempo, a mis maestros por transmitirme los conocimientos necesarios para afrontar una visa profesional y por ultimo a la universidad por permitirme representarla y ser parte de su gran familia.

1.2. Agradecimiento

Agradezco a mis padres por enseñarme el valor de las cosas, por sus consejos, por su preocupación y por acompañarme cuando la adversidad se hacía presente.

A Dios por darme la sabiduría y la inspiración para no abandonar este sueño que ahora se ha convertido en realidad.

Al Ing. Andrés Jiménez por aceptar ser director de este proyecto, su conocimiento y sabiduría aportan mucha calidad al trabajo.

1.3. Antecedentes

Existe una mínima cantidad de aplicaciones móviles orientadas al sector petrolero por lo que el desarrollo de una herramienta tecnológica aportaría de manera directa a las personas que desempeñan su trabajo en esta área, la generación de herramientas ingenieriles aumenta el reconocimiento internacional por lo que aporta tanto en el área publicitaria como en el área tecnológica.

El departamento de Innovación y Desarrollo de la empresa Sertecpet S.A ha recopilado un diccionario de fórmulas y algoritmos matemáticos los cuáles son la base de los cálculos para el aplicativo móvil

1.4. Justificación

La industria petrolera requiere de herramientas de cálculo para facilitar y efectivizar los trabajos de sus profesionales en las distintas operaciones que realizan diariamente.

SERTECPET S.A con su emprendimiento e innovación ve la necesidad de crear herramientas tecnológicas que sirvan de apoyo para el personal que labora en el ámbito petrolero, por lo que el desarrollo de este aplicativo móvil encaja perfectamente en el objetivo que se ha planteado dicha empresa.

La implementación de este aplicativo abarca la conversión de unidades más empleadas por la industria, así como cálculos básicos utilizados para encontrar volúmenes de tanques circulares y elípticos. La aplicación también implementará modelos IPR, los cuales ayudarán a determinar valores de producción de pozos petroleros usando variables definidas por los tres tipos de moedlo de cálculo IPR.

Este tipo de aplicaciones no se encuentran en gran porcentaje en tiendas de software, menos aún orientadas al sector petrolero, su potencial es alto e incluso podría ser utilizado con fines estudiantiles.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar una aplicación móvil de cálculos básicos utilizados por la industria petrolera.

1.5.2. Objetivos específicos

1.1.1..1.Documentar herramientas y metodologías a utilizar.

1.1.1..2.Documentar el alcance del requerimiento informático del proyecto y su relación con la solución específica a desarrollar.

1.1.1..3.Elaborar el diseño técnico de la solución a desarrollar.

1.1.1..4.Desarrollar la aplicación.

1.1.1..5.Realizar pruebas de funcionamiento.

2. Situación actual del desarrollo para móviles

2.1. Análisis de estado

El uso del internet a través del móvil aumenta a un ritmo frenético y, seguirá aumentando en los próximos años. El factor determinante ha sido la alta penetración de los Smartphone o teléfonos inteligentes. El nacimiento de este dispositivo replantea el acceso a la web, convirtiendo al teléfono móvil en un dispositivo de acceso rápido digital desde cualquier lugar. Mientras crece el número de personas que navegan a través de sus dispositivos móviles, pocas empresas tienen un claro conocimiento sobre lo que significa tener un web móvil, los beneficios que generan y como incorporar en sus estrategias de marketing y comunicación. (Mobile Marketing Association Spain, 2011)

Actualmente encontramos aplicaciones de todo tipo, forma y color, pero en los primeros teléfonos, estaban enfocadas en mejorar la productividad personal: se trataba de alarmas, calendarios, calculadoras y clientes de correo.



Ilustración 1 Evolución móviles

Hubo un cambio grande con el ingreso de iPhone al mercado, ya que con él se generaron nuevos modelos de negocio que hicieron de las aplicaciones algo rentables, tanto para desarrolladores como para los mercados de aplicaciones, como App Store, Google Play y Windows Phone Store.



Battery Life	iPhone 6 Plus	iPhone 6	iPhone 5s	iPhone 5c	iPhone 5	iPhone 4S
Talk Time	• Up to 24 hours on 3G	• Up to 14 hours on 3G	• Up to 10 hours on 3G	• Up to 10 hours on 3G	• Up to 8 hours on 3G	• Up to 8 hours on 3G
Standby time	• Up to 16 days (384 hours)	• Up to 10 days (250 hours)	• Up to 10 days (250 hours)	• Up to 10 days (250 hours)	• Up to 9 days (225 hours)	• Up to 8 days (200 hours)
Internet Use (3G)	• Up to 12 hours	• Up to 10 hours	• Up to 8 hours	• Up to 8 hours	• Up to 8 hours	• Up to 6 hours
Internet Use (LTE)	• Up to 12 hours	• Up to 10 hours	• Up to 10 hours	• Up to 10 hours	• Up to 8 hours	-
Internet Use (Wi-Fi)	• Up to 12 hours	• Up to 11 hours	• Up to 10 hours	• Up to 10 hours	• Up to 10 hours	• Up to 9 hours
Video Playback	• Up to 14 hours	• Up to 11 hours	• Up to 10 hours	• Up to 10 hours	• Up to 10 hours	• Up to 10 hours
Audio Playback	• Up to 80 hours	• Up to 50 hours	• Up to 40 hours	• Up to 40 hours	• Up to 40 hours	• Up to 40 hours

Ilustración 2 Evolución iPhone

Al mismo tiempo, también mejoraron las herramientas de las que disponían diseñadores y programadores para desarrollar apps, facilitando la tarea de producir una aplicación y lanzarla al mercado, incluso por cuenta propia.

Las aplicaciones móviles con el pasar del tiempo se han convertido en uno de los segmentos del marketing móvil que mayor crecimiento ha experimentado en los últimos años, las tiendas digitales de aplicaciones generaron un nuevo entorno de compra y venta creando nuevos nichos de negocios y oportunidades comerciales, estas tiendas han propiciado un aumento de consumo de aplicaciones entre los usuarios los cuales tienen la oportunidad de satisfacer sus necesidades a través de las mismas.

El ecosistema de móvil es la interacción directa entre agencias, desarrolladores, operadores, redes publicitarias, anunciantes y usuarios donde interactúan cooperando bajo un mismo entorno a fin de ofrecer soluciones al mercado mediante un dispositivo móvil que puede ser un Smartphone o un Tablet. (Mobile Marketing Association Spain, 2011)

El mercado de aplicaciones también ha influido en el mercado de “tabletas o tablets”, dispositivos móviles con una pantalla de mayores dimensiones y donde ciertas aplicaciones pueden consumirse con ciertas funcionalidades potenciadas, permitiendo entrar estos dispositivos en la industria y el sector empresarial.

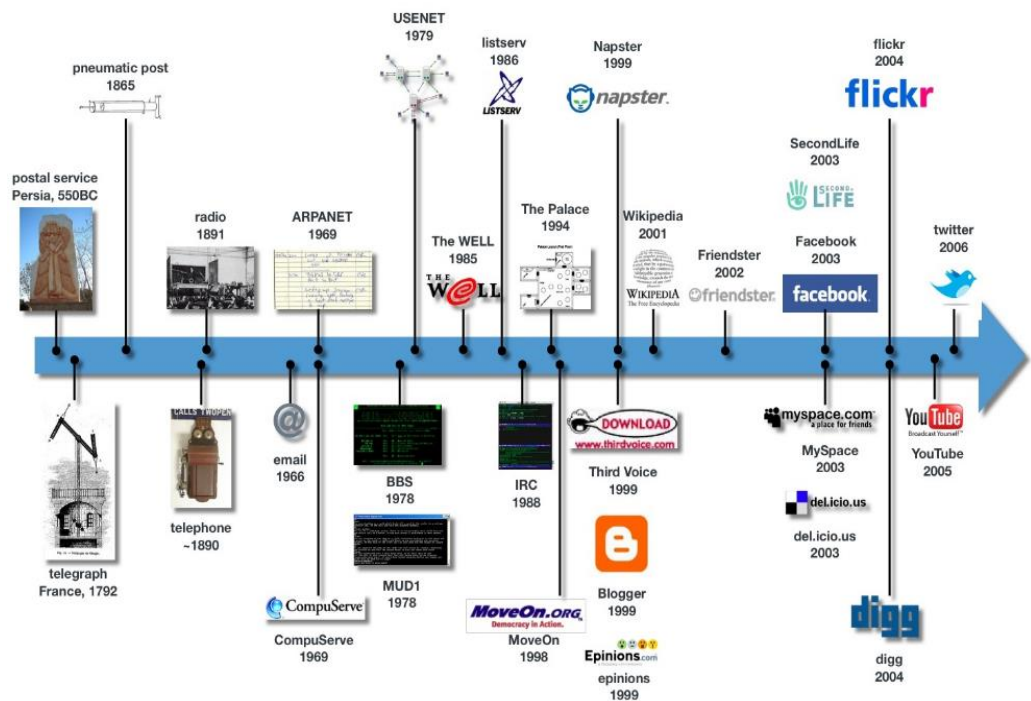
Todos estos dispositivos han causado un efecto dentro de varios sectores económicos, podemos hablar de los planes de navegación que ofertan las operadoras que se han visto beneficiadas por el auge de las apps, ya que la demanda de planes de datos ha subido sustancialmente, permitiendo generar nuevas estrategias de comercialización de teléfonos inteligentes.

Las funcionalidades como geolocalización, reconocimiento de voz, interacción, su carácter táctil, y muchas posibilidades más que permitan al usuario disfrutar de una navegación mucho más intuitiva. Todas estas herramientas permiten al usuario incrementar su capacidad de movimiento, mientras realizan negocios o disfrutan de su tiempo de ocio. (Mobile Marketing Association Spain, 2011)

Las aplicaciones móviles han producido una evolución obligada en el hardware de los dispositivos en los que se ejecutan debido a la competitividad del mercado, las más claras evoluciones en cuanto hardware están marcadas por la implementación del GPS, tarjetas aceleradoras gráficas para los usuarios

que prefieren utilizar sus dispositivos para jugar, sensores de proximidad, acelerómetro entre otros. Los constantes avances tecnológicos en hardware proveen al desarrollo de aplicaciones un gran marco de posibilidades para interactuar con los usuarios. Otro aspecto importante a mencionar entorno al hardware es el almacenamiento y memoria, estos avances hacen posible la ejecución de aplicaciones en paralelo permitiendo a los usuarios tener cuantas aplicaciones quieran y utilizar todas las que necesiten en un mismo dispositivo.

A través del uso de los dispositivos móviles se crean también nuevos hábitos de compra; y un objetivo principal de las aplicaciones también es la



venta de productos y servicios de una empresa. Este canal de venta puede ser complementario a otros canales on u offline, consiguiendo llegar a nuevos clientes que antes no habían sido abarcados o incluso pueden convertirse en un canal exclusivo si el servicio del producto es recomendable.

Ilustración 3Canal online

Muchas aplicaciones de gran éxito generan ingresos directos a sus desarrolladores o editores. Principalmente se diferencian dos modelos que podemos hacer referencia, por un lado, el planteamiento de comercializar espacios publicitarios en la propia aplicación que se distribuyen gratuitamente y, por otro lado, la venta de la aplicación en sí. Además, existen otros modelos como la venta de bienes digitales, “virtual currency”, la suscripción a contenidos o la compra de aplicaciones dentro de la aplicación.

Las funcionalidades que ofrecen las aplicaciones son muy diversas y cada día se inventan nuevos usos que activan o promueven nichos de mercado antes nunca imaginados, lo que nos hace pensar que nos encontramos ante un futuro digital por descubrir. (Mobile Marketing Association Spain, 2011)

Las empresas han encontrado a través de las aplicaciones móviles conseguir varios objetivos de comunicación y/o de negocio de una forma eficaz y eficiente. En la mayoría de los casos, la forma de alcanzarlos es única si se la compra con otros medios, ya que el uso específico de la tecnología y el uso que dan los usuarios a las aplicaciones abren posibilidades claramente diferenciadas. En cualquier caso, pueden aumentar la probabilidad de superar y aumentar las metas establecidas si el esfuerzo de conseguirlos está enmarcado en un planteamiento estratégico, tiene un enfoque o una presentación creativa y si está bien comunicada en todos los medios disponibles. (Mobile Marketing Association Spain, 2011)

Dentro de los objetivos de las empresas está que una aplicación sea generadora de notoriedad e imagen de marca. Esto a través de los contenidos, utilidades y/o las experiencias que ofrece la aplicación, con esto se puede impactar eficazmente a nuestro público objetivo generando notoriedad y una

imagen positiva hacia la marca. Este efecto se puede potenciar, ya que sabemos que el usuario se ha descargado la aplicación por lo que podemos deducir que esta ha generado interés hacia la misma. Si se consigue que el aplicativo aporte un cierto valor al usuario este fácilmente puede convertirse en objeto de recomendación, iniciando de esta forma un proceso viral. (Mobile Marketing Association Spain, 2011).

Las aplicaciones que han conseguido éxito suelen generar repercusiones mediáticas, lo cual aumenta su capacidad de generar notoriedad y resaltar la imagen de la marca.

La fidelización de usuarios y clientes es otro de los objetivos que suelen buscar las empresas, ya que las aplicaciones pueden establecer canales permanentes de comunicación lo cual las transforma en una herramienta idónea para conseguir la fidelización del cliente. La relación con los clientes se intensifica con la marca que como consecuencia se refleja un aumento del valor de la misma marca.

Todos estos fenómenos económicos y comerciales han causado una revolución en todos los campos, todos los ámbitos económicos y sociales han encontrado en las aplicaciones un importante medio ya sea de comunicación o como un utilitario de importante uso para las personas, todo esto genera un impacto global llevando los servicios y productos a un enlace obligatorio al sector móvil, el replanteamiento por parte de empresas en cuanto a su productividad y mercadeo, los móviles han marcado una nueva era tecnológica y han llegado para quedarse.

2.2. Tipos de aplicaciones móviles

“Una aplicación móvil, más comúnmente referido como una “app”, es un tipo de software de aplicación diseñado para ejecutarse en un dispositivo

móvil, como un ordenador teléfono inteligente o tableta. Las aplicaciones móviles con frecuencia sirven para proporcionar a los usuarios con servicios similares a los que se accede en las PC. Las aplicaciones son unidades de software en general pequeñas, individuales con función limitada”. (Techopedia, s.f.)

“Las aplicaciones móviles son un alejamiento de los sistemas integrados de software que generalmente se encuentran en los ordenadores personales. En cambio, cada aplicación proporciona una funcionalidad limitada y aislada, como un juego, calculadora o la navegación por Internet móvil. Aunque las aplicaciones pueden haber evitado la multitarea debido a los recursos limitados de hardware de los dispositivos móviles tempranos, su especificidad es ahora parte de su conveniencia, ya que permiten a los consumidores a recoger a mano lo que sus dispositivos son capaces de hacer”. (Techopedia, s.f.)

Una aplicación móvil también puede ser conocido como una aplicación, aplicación web, aplicación en línea, aplicación para teléfonos inteligentes.

2.2.1. Aplicaciones nativas

“Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si quieres que tu app esté disponible en todas las plataformas se deberán de crear varias apps con el lenguaje del sistema operativo seleccionado.” (LanceTalent, 2013)

Por ejemplo:

- Las apps para iOS se desarrollan con lenguaje Objective-C, Swift
- Las apps para Android se desarrollan con lenguaje Java
- Las apps en Windows Phone se desarrollan en .Net

El desarrollo de aplicaciones nativas tiene como ventaja el aprovechamiento de todas las características de hardware de los dispositivos tales como la cámara, GPS, agenda, almacenamiento entre otras. Esto tiene como resultado una mejora sustancial en cuanto a la experiencia de usuario o “UX” potenciando la aceptación de la aplicación. Entre las ventajas más considerables es que no requiere de una conexión a internet para funcionar.

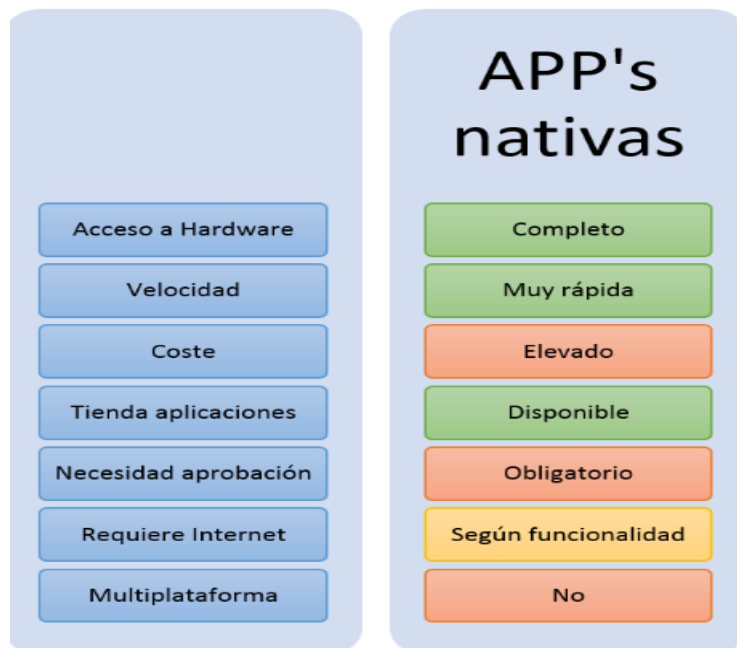


Ilustración 4App nativa

2.2.2. Aplicaciones móviles web

“Una aplicación web o web App es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, JavaScript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.” (LanceTalent, 2013)

Las aplicaciones móviles web se ejecutan en el navegador de dispositivos móviles a través de una URL, este tipo de aplicaciones suelen tener un costo más bajo ya que es la adaptación de un sitio web al navegador móvil, este tipo de aplicaciones no son dependientes de las diferentes tiendas móviles.

Una de las ventajas en cuanto a este tipo de aplicaciones es su distribución, no requiere de actualizaciones, el usuario siempre mantendrá la última versión. Este tipo de aplicaciones son totalmente dependientes de una conexión a internet y su velocidad de funcionamiento depende de la velocidad de internet con la que cuente cada usuario.

Este tipo de aplicaciones no pueden tener acceso al hardware del dispositivo por lo que su funcionamiento está limitado a lo que el navegador web pueda ejecutar, este aspecto abre una nueva desventaja, no todos los dispositivos móviles usan el mismo navegador y la variedad de navegadores crean un enorme problema en cuanto a compatibilidad de los componentes web.

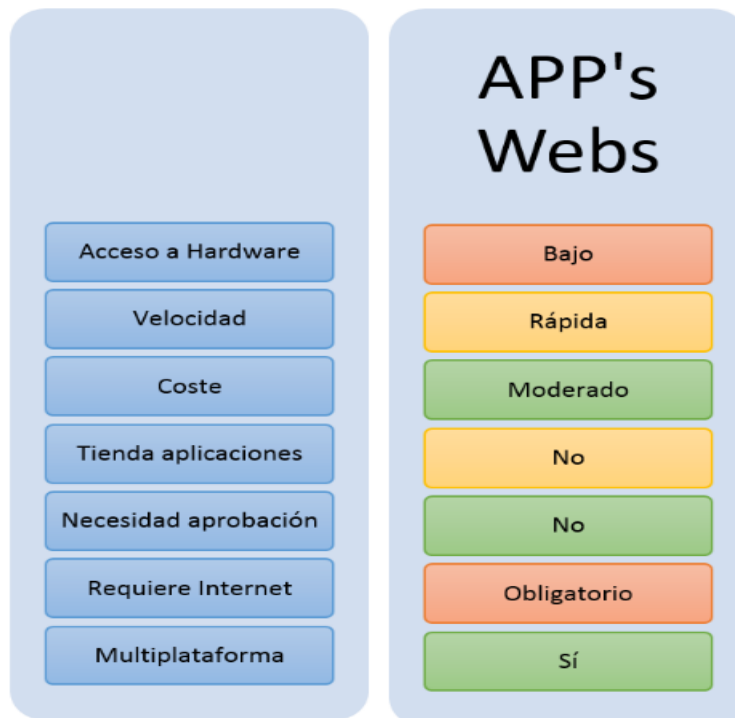


Ilustración 5App web

2.2.3. Aplicaciones híbridas

El enfoque híbrido combinan desarrollo nativo con tecnología Web. Usando este enfoque, los desarrolladores escriben gran parte de su aplicación en tecnologías Web para múltiples plataformas y mantienen el acceso directo a APIs nativas cuando lo necesitan. (IBM, 2012)

La porción de la aplicación emplea APIs de sistemas operativos para crear un motor de búsqueda HTML incorporado que funcione como un puente entre el navegador y las APIs del dispositivo. (IBM, 2012)

Este camino generado permite que una aplicación híbrida tenga la posibilidad de aprovechar todas las características que ofrecen los dispositivos móviles, incluyendo hardware. Existen una diversidad que ayudan con la generación de este tipo de aplicaciones, estas herramientas cuentan con bibliotecas que proveen una interfaz JavaScript uniforme para funcionalidades de dispositivos seleccionados que son todos iguales en los sistemas operativos.

La parte nativa de estas aplicaciones se pueden desarrollar de manera independiente, sin embargo, la mayoría d herramientas ofrecen este tipo de contenedor nativo como parte de su producto, lo que rinda al desarrollador tener la facilidad de implementar aplicaciones avanzadas utilizando lenguaje web.

Por ejemplo, Ionic es una herramienta que proporciona funcionalidades necesarias para el desarrollo de aplicaciones híbridas basándose en lenguajes web como JavaScript y HTML con un fuerte soporte en Angular, este Framework genera soluciones a partir de un solo desarrollo para IOS y Android.

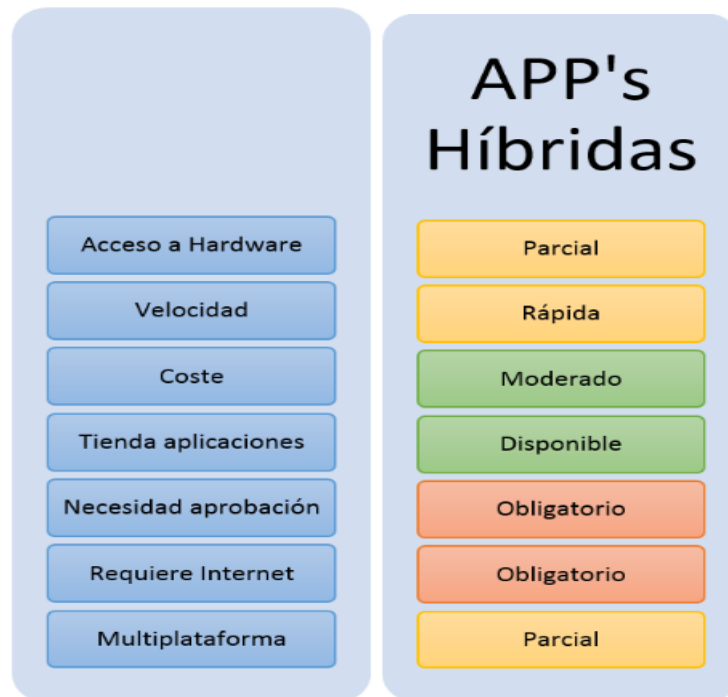


Ilustración 6App híbridas

2.2.4. Análisis de ventajas y desventajas

- Los costes de desarrollo son menores que los de las apps nativas ya que es un solo producto distribuible para otras plataformas.
- El funcionamiento y la experiencia de usuario a nivel de rendimiento es igual que una app nativa.
- La misma base de código nos permite desplegar la app en Android, iOS y Microsoft Windows (entre otros).
- Utilizando las técnicas adecuadas, 100% del código del proyecto es multiplataforma.
- Permite distribución a través de las diferentes tiendas de aplicaciones por lo que nos podemos beneficiar de sus pasarelas de pago y de su difusión.
- Permite acceso completo a la funcionalidad y servicios del dispositivo (GPS, cámara, etc.)

- Tiene una amplia comunidad de desarrolladores con lo que tiene buen soporte técnico y existen muchas librerías que se pueden reutilizar para cubrir distintos casos de uso.
- El mantenimiento es más sencillo ya que el código fuente es el mismo para todo nuestro proyecto.

2.3. Aplicaciones empresariales

Las aplicaciones móviles permiten crear y mantener una relación con el cliente. Es la mejor herramienta de fidelización, creando una valiosa interacción que posiciona a la marca y ofrece al mismo tiempo una imagen de innovación. Permite realizar ofertas especiales a los clientes, obtener su opinión, facilitar su compra tanto online como física y motivar su recomendación. (Altiria, n.d.)

Asimismo, las aplicaciones empresariales son usadas para ampliar el mercado a fin de obtener un acceso de comercialización y servicios, logrando que estos funcionen como una nueva estrategia de marketing en la empresa, del objetivo parte el desarrollo de las aplicaciones móviles por lo que es necesario que no existan dudas.

Las aplicaciones dependen de ciertas particularidades para ser desarrolladas, ya que no es lo mismo si se trata de; una empresa o una multinacional, el sector a la que va dirigida, si lo que se busca es fidelización o crear un nuevo canal de negocio.

Nuevas corrientes aumentaran la presión sobre las compañías para desarrollar una mayor variedad de aplicaciones móviles en plazos más cortos. El rápido desarrollo, despliegue y mantenimiento de aplicaciones móviles

corporativas será un reto inexcusable para las empresas debido a la creciente demanda de las apps en el mercado. (Silicon, 2015)

Las herramientas móviles y las estrategias técnicas dependerán de la capacidad de los desarrolladores de aplicaciones móviles porque sus departamentos están encargados de la innovación para sus empresas y así satisfacer el contacto con los usuarios, poder encontrar nuevos y mejores clientes y brindar disponibilidad de los recursos de las apps de la empresa las 24 horas del día. De esta forma las empresas mejoran la calidad de actividades comerciales que ofrecen obteniendo una recompensa por estos servicios. Asimismo, los programadores deben optar por priorizar sus actividades por el exceso de trabajo que las aplicaciones generan, ya que son la clave para la evolución de los proyectos optimizando las habilidades móviles brindadas.

2.4. Metodología de diseño

El diseño de aplicaciones móviles, comprende un espectro muy grande en cuanto a factores a considerar, se la considera como una de las etapas más críticas en la creación de una aplicación esto debido al impacto que genera el diseño en los cliente, en muchos de los casos un buen diseño llega a ser un factor determinante para el usuario final, por la frecuencia de uso que esperan tener las aplicaciones deben ser lo suficientemente amigables y fáciles de usar para que el usuario no quiera probar con otras en un mar de opciones. (Vittone, 2013)

La interfaz de una aplicación es como la ropa que viste para salir a la calle. Es también la capa que hay entre el usuario y el corazón funcional de la app, el lugar donde nacen las interacciones. (Vittone, 2013)

En la etapa de diseño es importante interpretar la personalidad de cada uno de los sistemas operativos ya que uno de ellos tiene una visión y estilo diferente, por lo que realizar un diseño que mantenga concordancia con el del sistema operativo es un factor a considerar.

El diseño está compuesto especialmente por componentes utilizados durante la ejecución de la aplicación como lo son los botones, títulos, pestañas entre otros, cada uno de estos componentes aporta crucialmente a la interacción con el usuario, esto va muy de la mano con la experiencia de usuario.

La combinación de estilos de los diferentes sistemas operativos es muy aplicada a menudo ya que se busca encontrar un balance adecuado, el tipo de aplicación incide directamente en el diseño, pueden llegar a requerir controles muy personalizados u otras que simplemente ocupan elementos nativos de cada sistema operativo.

Una aplicación entre otras cosas es un medio de comunicación que forma parte de un sistema y es una oportunidad para que las marcas o las empresas se den a conocer a través de las distintas pantallas. Las diferentes pantallas de la aplicación tratan de reflejar conceptos manejados por cada empresa, por lo general son utilizadas las pantallas introductoras para este fin.

2.4.1. Metodologías existentes

METODO ADAPTATIVO	METODO PREDICTIVO
<ul style="list-style-type: none"> • Permite adaptarse con facilidad a los cambios. 	<ul style="list-style-type: none"> • La prioridad es estudiar y proyectar a detalle el futuro.
<ul style="list-style-type: none"> • La organización es flexible a diversos cambios para llegar a su camino. 	<ul style="list-style-type: none"> • Los riesgos no pasan desapercibidos siempre son bien estudiados.
<ul style="list-style-type: none"> • Es complejo describir la adaptación de cambios exactos a futuro. 	<ul style="list-style-type: none"> • Mientras el proyecto está en progreso se planean las características y las tareas específicas.

<ul style="list-style-type: none"> Las tareas próximas no pueden ser notificadas, debido a la dificultad que tiene para detallar puntualmente lo que está por ocurrir. 	<ul style="list-style-type: none"> Si el proyecto falla es debido a que la fase anterior tuvo problemas, por lo que cambiar de trayectoria sería un gran inconveniente.
	<ul style="list-style-type: none"> Solo las modificaciones más acertadas son considerados en sus tableros.
METODO ITERATIVO	METODO CASCADA
<ul style="list-style-type: none"> Su objetivo es la calidad y las pruebas por lo que suelen fusionarse simultáneamente en el desarrollo. 	<ul style="list-style-type: none"> Cuenta con dos etapas alejadas denominadas de pruebas y de construcción.
<ul style="list-style-type: none"> Las pruebas son elaboradas en pequeños fragmentos de software. 	
<ul style="list-style-type: none"> Los consumidores pueden manipular y valorar las partes que van saliendo. 	

2.4.2. Agile modeling

Se recurre al modelo ágil para desplegar un punto de vista en el desarrollo de software, estableciendo modelos adelantados y amplios de los códigos. Por lo general se adopta un enfoque que consta de etapas para el desarrollo, los más usuales manifiestan que es frecuente obtener un camino iterativo. Por lo tanto, crean modeladores extensos antes de escribir un código, generan modelos ágiles que son lo suficientemente buenos para impulsar el desarrollo general. Asimismo, es considerado como estrategia indispensable para escalar el desarrollo ágil de software más allá de la pre ubicación, tomando en cuenta la secuencia ya establecida.

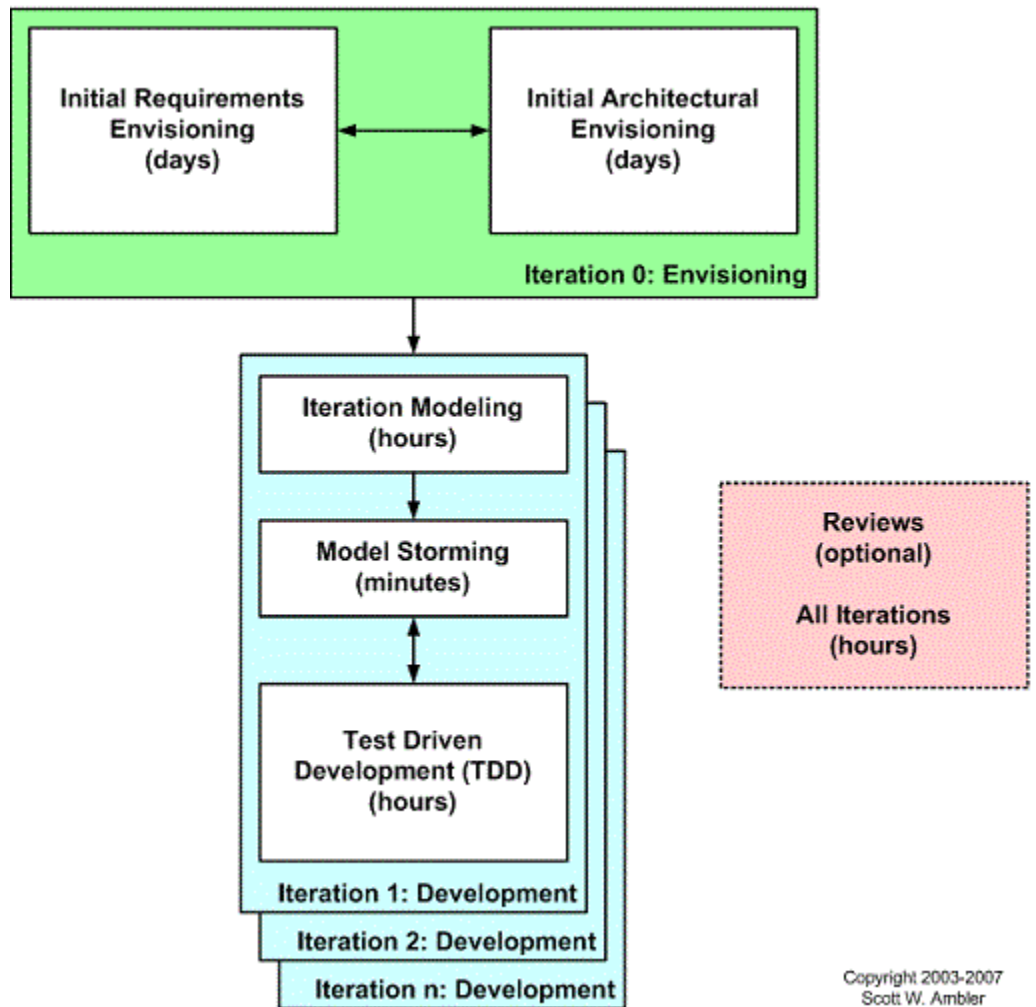
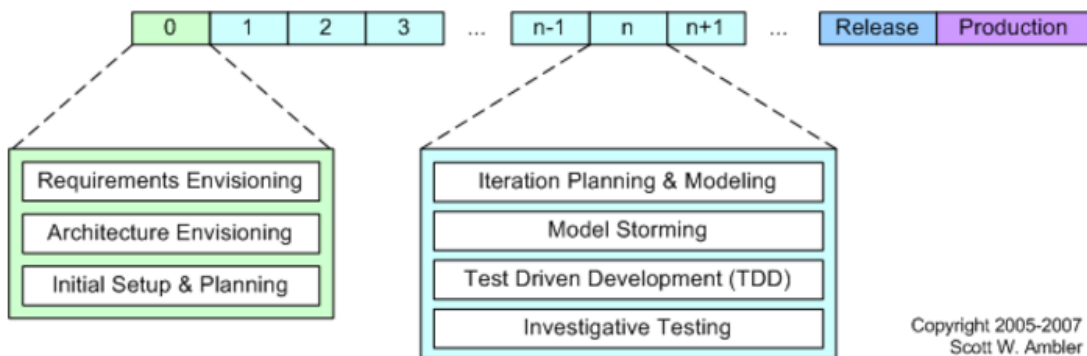


Ilustración 7Ciclo de vida

El grafico representa cada actividad en la vida de un sistema nuevo, al igual, incluye subsistemas que se generan en el inicio y en la arquitectura del sistema, en los otros cuadros se detallan las actividades que ocurren en la iteración esto se da en un tiempo específico para cada cuadro. La arquitectura promueve la trayectoria técnica factible para generar el proyecto y proporcionar suficiente información para organizarla. Por otra parte, la iteración se encuentra en constante evolución ya que se debe planificar el trabajo a realizar.



*Ilustración 8*Trancurso

También se observa el proceso del desarrollo del método ágil por medio del software en el ciclo de vida ya que comienza con el modelo inicial y continúa produciéndose en cada módulo siguiente.

El esfuerzo se realiza durante la primera semana de un proyecto, cuyo objetivo es identificar el alcance de su sistema y una arquitectura probable para hacer frente. Para ello se ejecuta ambas cosas de alto nivel de modelado de requisitos y la arquitectura de modelado de alto nivel. El objetivo no es escribir las especificaciones detalladas, eso resulta muy arriesgado, en la práctica, pero en vez de explorar las necesidades es preferible llegar a una estrategia global para su proyecto. Para los proyectos cortos (quizás varias semanas de duración) que puede hacer este trabajo en las primeras horas y para proyectos largos (quizás de doce meses o más), puede decidir invertir dos semanas en este esfuerzo. (Ambler, 2003-2012)

Por lo tanto, el modelado inicial es indispensable para guiar el proyecto a un alto nivel, su objetivo es causar una buena impresión del software que se va a desarrollar, para alcanzarlo es necesario de algunos días para conseguir los requisitos, también el alcance de lo que se va a ejecutar, finalmente identificar las características de negocios y la relación que existe entre un modelo de usuario y el uso que se proporcionará.

2.5. Metodologías de desarrollo

La etapa del desarrollo necesita de ciertos elementos claves para su inicio que estén muy bien definidos en fases anteriores como la finalidad de la app, los destinatarios y su forma de utilizar la aplicación y su funcionalidad básica, el desarrollador debe asegurarse de que todos los elementos mencionados sean totalmente completos para empezar la fase de desarrollo.

En el mundo del desarrollo móvil existen una variedad de herramientas disponibles, en la mayoría de los casos puestas a disposición por los propios creadores de sistemas operativos denominados SDKS, existen herramientas externas que prestan funcionalidades como servicio las cuales permiten reducir tiempo y costos en la mayoría de los casos permitiendo tener un desarrollo más amplio sin mayores costos.

Los SDKS de cada sistema operativo cuentan con una fuerte documentación las cuales permiten tener una fuente de consulta veraz para los desarrolladores, los diferentes tipos de aplicaciones móviles utilizan diferentes lenguajes de programación, ya que los SDKS de cada sistema operativo está desarrollado en un lenguaje diferente, el esquema híbrido y móvil únicamente comparten el mismo lenguaje ya que su arquitectura es netamente web por lo que los dos utilizan lenguajes como HTML, Javascript y CSS.

El desarrollo de aplicaciones híbridas tiene la finalidad de realizar un único desarrollo que pueda ser distribuido posteriormente a otras plataformas, para esto los frameworks de este tipo de aplicaciones implementan funcionalidades que permiten conectar el ambiente web junto con el hardware del dispositivo, este tipo de desarrollos se ejecutan en un servidor web durante su construcción o se pueden utilizar emuladores de los diferentes sistemas existentes.

2.5.1. Metodologías existentes

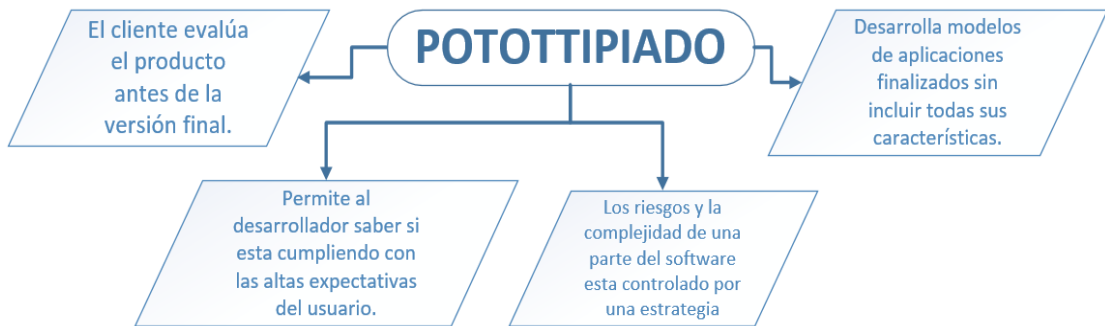


Ilustración 9 Modelo Prototipado

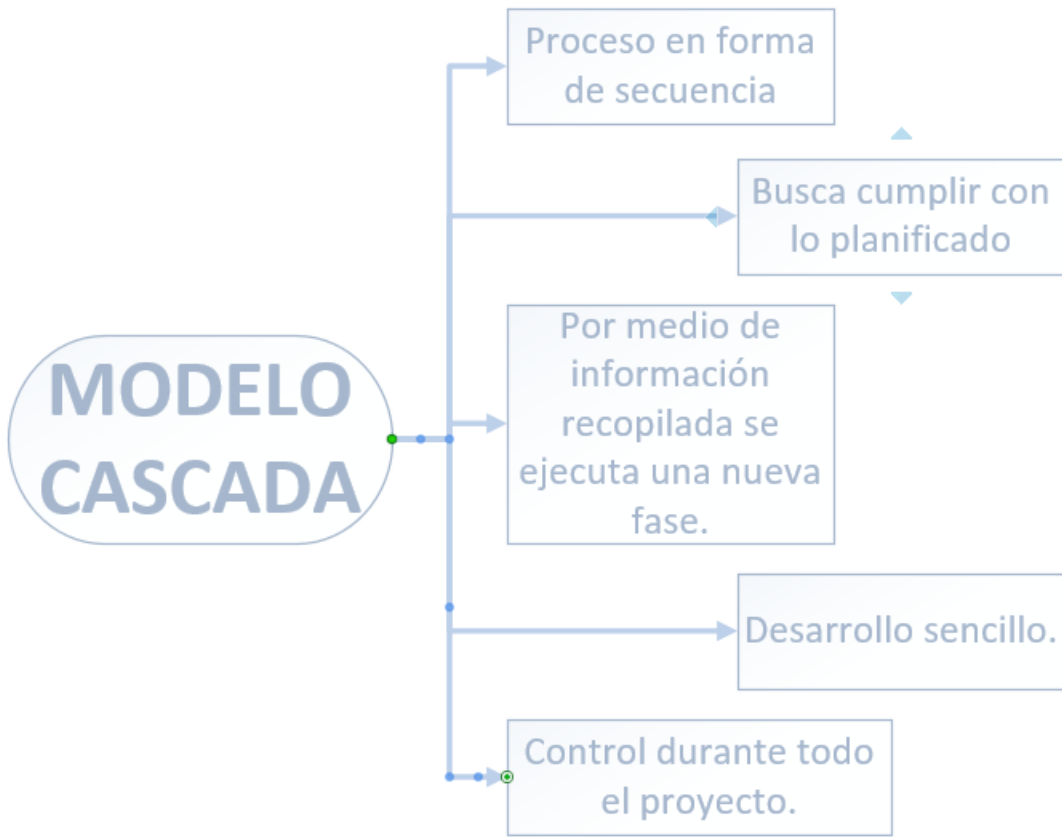


Ilustración 11 Modelo Cascada

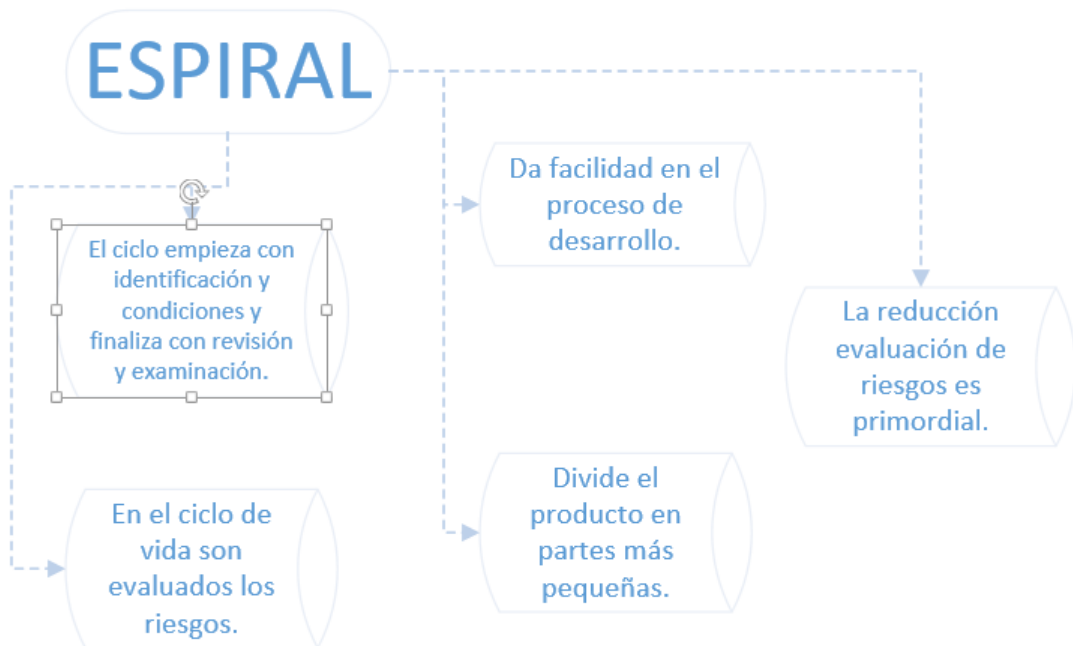


Ilustración 10 Modelo Espiral

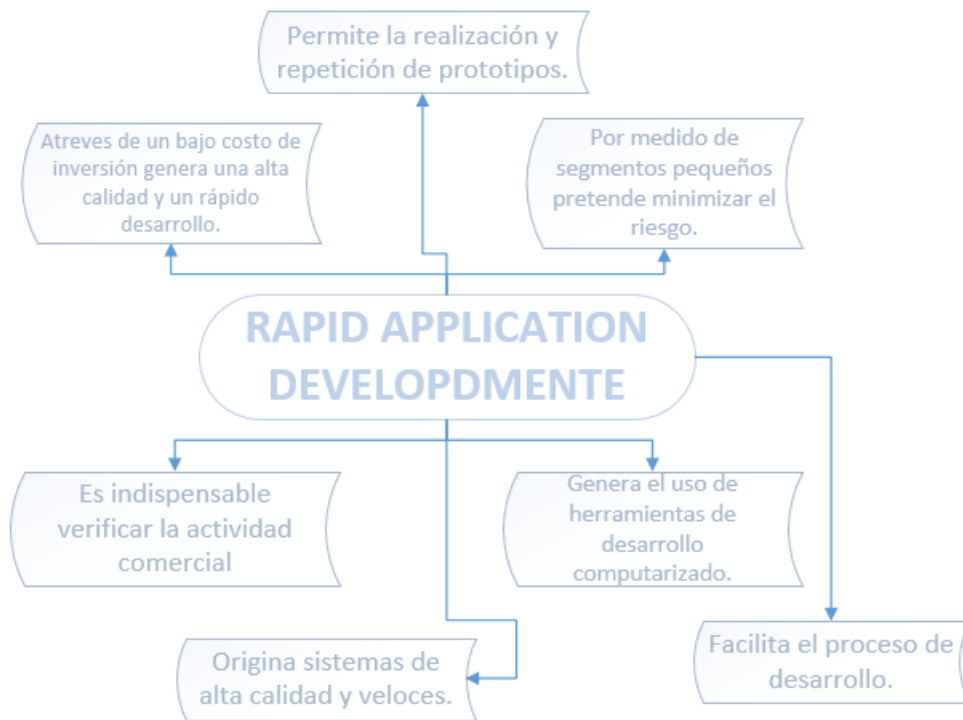


Ilustración 12 Modelo RAD

2.5.2. Análisis de ventajas y desventajas

- Modelo en cascada

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • Debido a la fácil implementación en productos que están por culminar disminuye el capital para generarlo, asimismo, para que marche eficazmente. 	<ul style="list-style-type: none"> • Mientras el software no se encuentre acabado durante el proceso de prueba no puede ser operado en el proceso de creación de software, por lo que se convierte en un proceso lento.
<ul style="list-style-type: none"> • Efectuarlo y comprenderlo es sencillo. 	<ul style="list-style-type: none"> • En la práctica, pocas veces un plan sigue una continuidad lineal.
<ul style="list-style-type: none"> • Implementado con 	<ul style="list-style-type: none"> • Si se encuentra un error en la

frecuencia, destacado por generar una mayor cantidad de documentación.	fase de prueba se debe rediseñar, incrementando el valor de desarrollo.
--	---

- Metodología prototipado

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • El usuario conoce las características del software, pero no los requisitos para ser desarrollado. 	<ul style="list-style-type: none"> • El usuario suele pensar que se encuentra frente al producto finalizado.
<ul style="list-style-type: none"> • Su enfoque se basa en la sinergia entre el producto y el usuario. 	<ul style="list-style-type: none"> • Por lo general se deja de lado la razón por la que se comenzó a realizar el software.
<ul style="list-style-type: none"> • Es independiente el proceso de construcción del modelo de prototipos. 	

- Metodología incremental

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • No es indispensable que el sistema esté finalizado para ser ejecutado. 	<ul style="list-style-type: none"> • Al generarlo prematuramente el sistema resulta aburrido, simple para el usuario.
<ul style="list-style-type: none"> • Permite dividir el proyecto por bloques según su dificultad. 	<ul style="list-style-type: none"> • Se pierde la objetividad del sistema al utilizar las líneas de código muy cargadas.
<ul style="list-style-type: none"> • Las probabilidades de riesgo son nulas 	<ul style="list-style-type: none"> • Por lo general el cliente no cuenta con la disponibilidad para estar durante la generación del proyecto.
<ul style="list-style-type: none"> • Las características del producto son precisas y estables. 	<ul style="list-style-type: none"> • El progreso del sistema no es independiente, sino en mutuo acuerdo con el cliente.
	<ul style="list-style-type: none"> • Agentes externos pueden vulnerar con facilidad el correcto funcionamiento de programa.

- Metodología espiral

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • El riesgo del proyecto es mínimo. 	<ul style="list-style-type: none"> • Se tarda en la realización del desarrollo del sistema.
<ul style="list-style-type: none"> • Genera características y objetivos de calidad. 	<ul style="list-style-type: none"> • Crear con este modelo suele ser imposible
<ul style="list-style-type: none"> • Tanto el desarrollo como el mantenimiento van de la mano. 	<ul style="list-style-type: none"> • Sin un presupuesto no puede comenzar a funcionar.
<ul style="list-style-type: none"> • El ciclo de vida no es estricto por lo que se puede realizar mejoras en cualquier etapa 	<ul style="list-style-type: none"> • Sin experiencia es imposible reconocer los riesgos.

- Metodología rapid application development

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • La participación comercial es fundamental. 	<ul style="list-style-type: none"> • La ingeniería tecnológica es de menor importancia.
<ul style="list-style-type: none"> • Promueve la información necesaria para el mantenimiento a futuro. 	<ul style="list-style-type: none"> • Necesita colaboración activa de los usuarios.
<ul style="list-style-type: none"> • Prioriza la calidad y la rapidez de los sistemas. 	<ul style="list-style-type: none"> • Pretende minimizar los riesgos del proyecto.

2.5.3. Prototyping

El prototipado permite desarrollar modelos de aplicaciones de software que permiten ver la funcionalidad básica de la misma, sin necesariamente incluir toda la lógica o características del modelo terminado. El prototipado permite al cliente evaluar en forma temprana el producto, e interactuar con los diseñadores y desarrolladores para saber si se está cumpliendo con las expectativas y las funcionalidades acordadas. (Fundación Wikimedia, 17)

Los prototipos no poseen la funcionalidad total del sistema, pero si condensa la idea principal del mismo, Paso a Paso crece su funcionalidad, y maneja un alto grado de participación del usuario. (Fundación Wikimedia, 17)

Prototyping conduce al desarrollo de técnicas que representan varios métodos de soluciones hasta llegar a la más eficiente con el fin de lograr resolver un problema específico. Necesitan ser ajustados hasta encontrar la que se encargue de cubrir todas características que genera el software, para así ejecutar pruebas que obedezcan al modelo de prototipo que se use para el desarrollo del dispositivo.

Prototyping está encargado de cumplir varias funciones entre estas tenemos el desarrollar, modelar y evaluar diseños para usarlos mediante pruebas, es decir que no tiene necesidad de estar culminado para ser ejecutado, lo que permite crear un sin número de variantes en nuestra aplicación. Asimismo, este admite evaluar nuevos diseños desechando los anteriores.

"Escribir un programa didáctico es como tener una aventura: generalmente conocemos el punto de partida, más o menos sabemos dónde queremos ir, pero desconocemos con exactitud lo que pasará por el camino" (Jean Michel Lefèvre)

Prototyping es un tipo de desarrollo paulatino, por lo que es necesario edificarlo en un corto periodo y sin recurrir a tantos recursos. Cuenta con un diseño rápido debido a las características con las que cuenta el software y que solo el usuario las podrá ver, la iteración se ejecuta al hacer las modificaciones en la última etapa cumpliendo con las necesidades del consumidor en un periodo de tiempo pequeño.

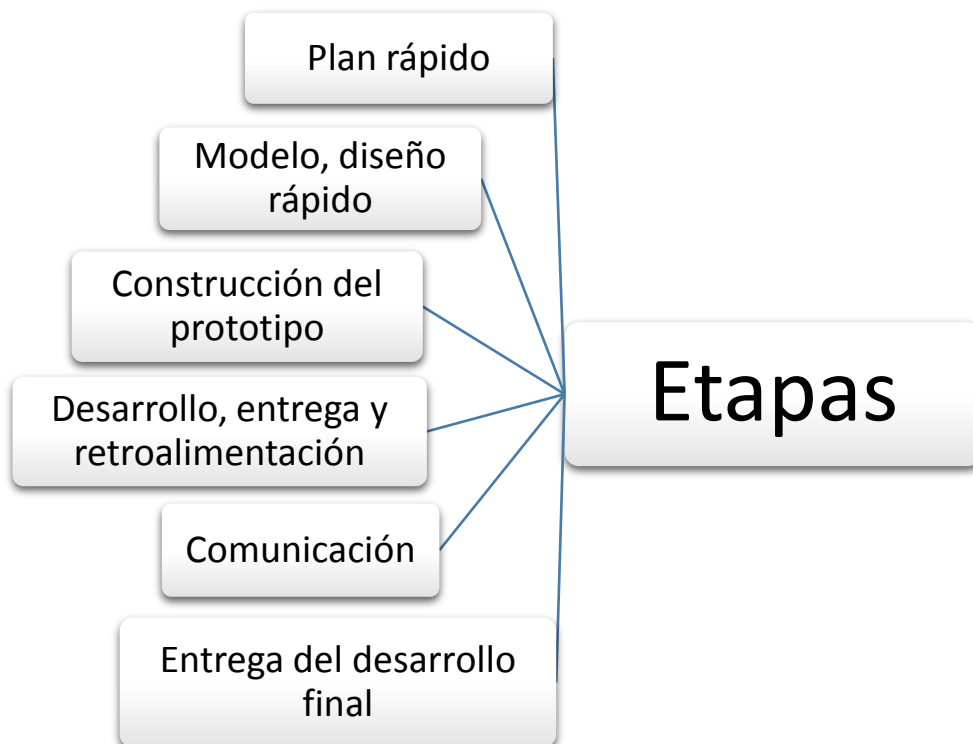


Ilustración 13 Etapas prototipado

Prototipado es el más eficiente porque no depende de un proceso exacto, además, está encargado del requerimiento de desarrollo de software encaminado a la calidad. Igualmente, la relación que tiene el usuario con el desarrollador le permite cumplir con sus disposiciones y de esta manera deja como resultado una aplicación que respeta las solicitudes del consumidor final.

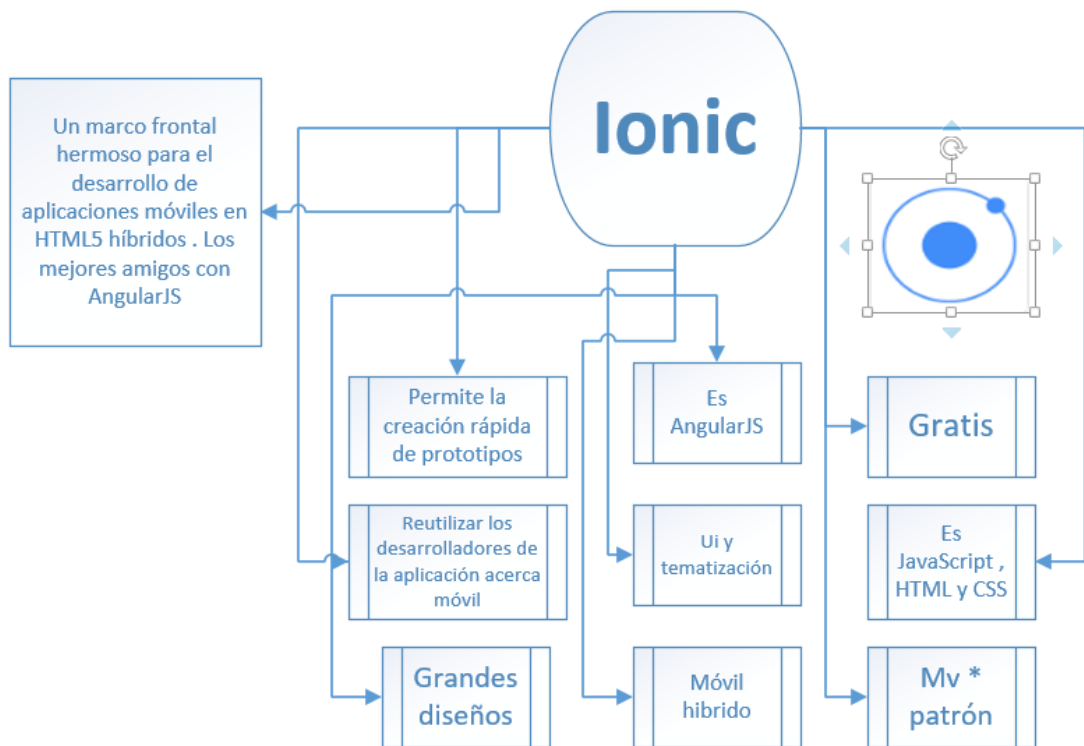
- Análisis de ventajas y desventajas de prototyping

Ventajas:	Desventajas:
<ul style="list-style-type: none"> • Al usarlo se establece una mejor comunicación entre los usuarios y el desarrollador, aumentando la aceptación del nuevo sistema 	<ul style="list-style-type: none"> • Los usuarios suelen pensar que está terminado al ver que funciona, cuando en realidad se está desarrollarse el producto.

y así reducir la incertidumbre del riesgo.	
<ul style="list-style-type: none"> • Son software montados en poco tiempo, el costo es mínimo y mientras no esté aprobado no empieza el desarrollo final del software. 	<ul style="list-style-type: none"> • Al contar con dos clases de prototipos se vuelven indispensables, no funciona el uno sin el otro y se los conoce como desechable y evolucionario.
<ul style="list-style-type: none"> • Permite a los usuarios tener una vista previa de una parte del software. 	<ul style="list-style-type: none"> • Si al usuario no le agrada lo que ve, debe ser rectificadada hasta que el interesado quede satisfecho.

2.6. HERRAMIENTAS DE DESARROLLO

2.6.1. Herramientas disponibles



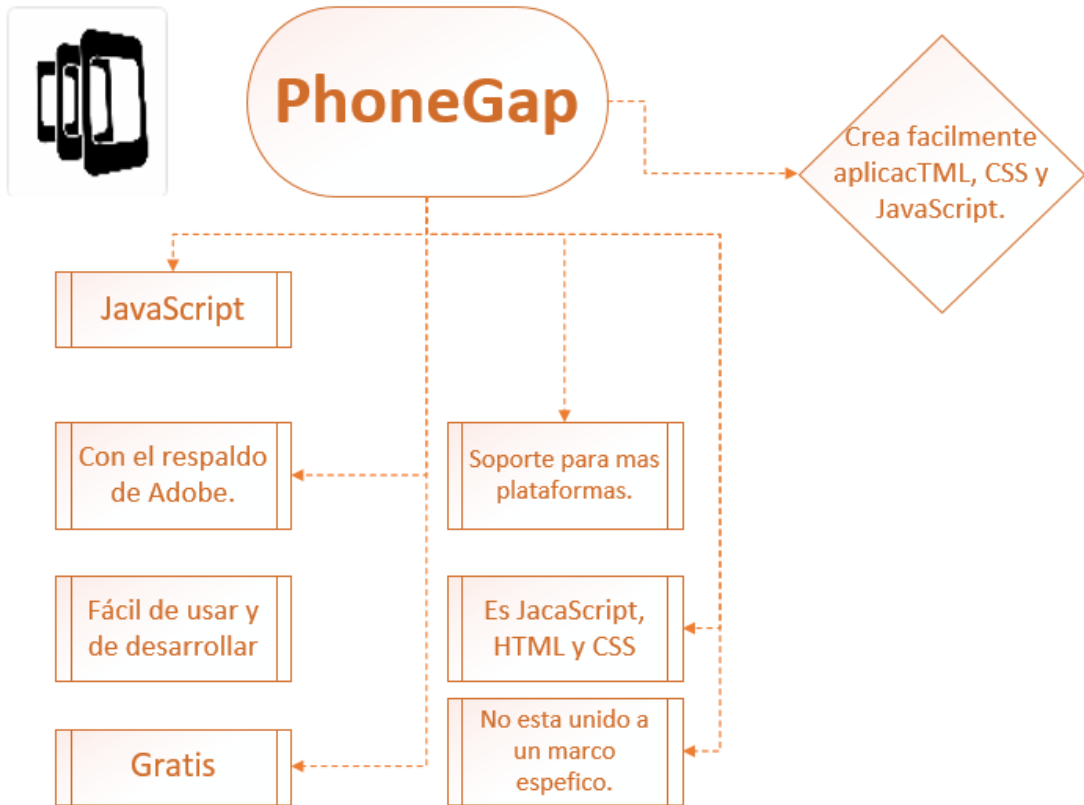


Ilustración 14PhoneGap

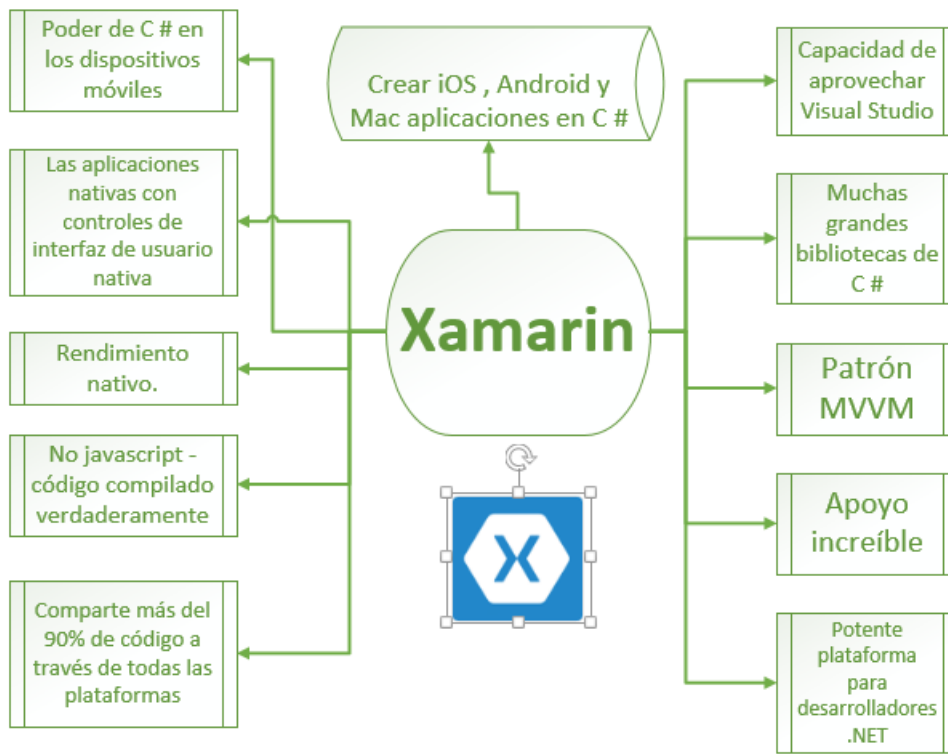


Ilustración 15Xamarin

P
ara el
desarr
ollo
de
aplica
cione
s
nativa
s
existe
n
varios
marco

s de desarrollo entre los cuales los mencionados anteriormente destacan sobre el resto. Ionic nos ofrece enormes ventajas para el desarrollo de este proyecto ya que está orientado al prototipado de aplicaciones. Esta ventaja de Ionic permite apegarnos adecuadamente a nuestra metodología de desarrollo y ser mucho más eficientes al momento de mostrar resultados.

PhoneGap como Xamarin son marcos de desarrollo respaldados por grandes empresas, Adobe y Microsoft respectivamente, cada una de ellas tiene el mismo propósito que Ionic que es la generación de aplicaciones móviles híbridas. Cada una de estas herramientas buscan impulsar el desarrollo de este tipo de aplicaciones, complementando sus desarrollos con otras herramientas que proveen las empresas a las que pertenecen.

Desde el punto de vista del desarrollador Ionic mantiene un marco de referencia más definido utilizando Angular Js, esto permite tener un desarrollo más controlado, de esta característica carece PhoneGap ya que en su marco de desarrollo está vierto a la implementación de cualquier framework JavaScript. Xamarin por su lado ofrece este marco de referencia utilizando librerías C#, para lo que los desarrolladores deben tener conocimiento en plataforma .net, esto de cierto modo marca una limitante ya que incluimos el manejo de otro lenguaje al momento del desarrollo.

Una de las ventajas que pondera Ionic es la reutilización del código para otras aplicaciones ya sean móviles o web, este es un punto que el departamento considero vital al momento de seleccionar una plataforma.

2.6.2. Ionic app framework

Ionic es un marco de desarrollo de aplicaciones móviles HTML5 dirigida a la creación de aplicaciones móviles híbridas, Las cuales son esencialmente pequeños sitios web que se ejecutan en el navegador de la aplicación y tienen acceso a la capa de plataforma nativa. Las aplicaciones híbridas tienen muchas ventajas sobre las aplicaciones nativas puras, específicamente en términos de soporte de la plataforma, la velocidad de desarrollo, y el acceso al código de terceras partes.(Ionic)

Ionic implementa un gran marco en cuanto a interfaz y manejo de experiencia de usuario, lo que permiten el desarrollo de aplicaciones visualmente agradables y con una interacción poco complicada para el usuario.

A diferencia de un marco de respuesta, Ionic implementa controles de interfaz muy móviles, nativos de estilo y diseño que normalmente se obtendrían con un SDK nativo en iOS o Android que no existían antes en la web. Ionic también le da gran alcance para construir aplicaciones móviles que eclipsan los marcos de desarrollo de HTML5 existentes.

Ionic es un marco HTML 5, que necesita una envoltura nativa como Córdoba o PhoneGap con el fin de funcionar como una aplicación nativa. Se recomienda encarecidamente utilizar Cordova el cual es el más adecuado para sus aplicaciones y las herramientas que utiliza Ionic por detrás. (Ionic)

Quienes están familiarizados con el desarrollo web encuentran sencilla la estructura de una aplicación Ionic. En su esencia, es sólo una página web que se ejecuta en una cáscara de aplicación nativa. Eso significa que podemos utilizar cualquier tipo de HTML, CSS y JavaScript que queramos. La única diferencia es que en lugar de crear un sitio web que los demás enlazar, estamos construyendo una experiencia de aplicación autónoma.

Todo el desarrollo de una aplicación Ionic será escrita en HTML, JavaScript y CSS. Existe también la posibilidad de realizar implementaciones en la capa nativa con plugins personalizados de Córdoba o código nativo, pero no son necesarios para obtener una gran aplicación.

Iónica también utiliza AngularJS para una gran parte de la funcionalidad básica del marco. Si bien todavía se puede utilizar iónico con sólo la parte de CSS, se recomienda invertir en angular, ya que es una de las mejores maneras de construir aplicaciones basadas en navegadores actualmente. (Ionic)

- *Requerimientos para el desarrollo*

Para el desarrollo de aplicaciones Ionic se debe tomar en cuenta unos requisitos mínimos para la construcción de su aplicación con la versión actual de la base Ionic. Ionic se dirige a dispositivos iPhone y Android (en la actualidad). Ionic soporta iOS 7+, y Android 4.1 o superior. Sin embargo, puesto que hay una gran cantidad de diferentes dispositivos Android, es posible que ciertas aplicaciones en ciertos dispositivos podrían no funcionar. Ionic se mantiene en actualización constante y su compatibilidad con la mayoría de dispositivos constantemente crece.

Se puede usar cualquier sistema operativo para el desarrollo de aplicaciones Ionic. Ionic se ha desarrollado en varias ocasiones en Mac OS X, Linux y Windows. Sin embargo, se deben tomar en cuenta los comando a ejecutarse en consola únicamente, para el desarrollo de aplicaciones IOS se necesita una máquina que ejecute Mac OS X para su futura conversión para este tipo de dispositivos.

Si se desarrolla en un ambiente Windows es necesario instalar GIT, Node Js, Apache Crodova

Si estamos corriendo Ubuntu 64-bits, va a ser necesario que instalemos las librerías de 32-bits porque Android corre solo en 32-bits hasta el momento.

Usuarios de Windows que deseen desarrollar en Android: Tendrán que asegurarse de instalar y configurar los siguientes programas.

- Java JDK

- Apache Ant
- Android SDK

Proyecto Ionics

Para realizar un directorio necesitamos comandos de esta forma se ira creando, ejecutando hasta ver la estructura final. Estos comandos se irían generar de tal forma que el anterior consiga una estructura parecida a la que podemos ver en la siguiente imagen: se irían generar

```
$ cd todo && ls
| bower.json // dependencias de bower
| config.xml // configuración de cordova
| gulpfile.js // tareas de gulp
| hooks
| ionic.project // configuración de ionic
| package.json // dependencias de nodejs
| platforms // compilaciones de iOS/Android estarán en este dir
| plugins // Plugins de cordova y ionic
| scss // código scss, que compilara a css y lo guardara en www/css/
| www // la aplicación propiamente dicha(codigo javascript, css, html, imagenes, etc)
```

Ilustración 16 Archivos proyecto

```
$ ionic build ios
$ ionic emulate ios
```

La forma en la que sabemos que es correcto el funcionamiento del proyecto es por medio de una prueba, compilando y corriendo el proyecto,

(para iOS deberemos correr los siguientes comandos, pero si estamos en Android solo reemplazaremos iOS por Android en el comando): (Ionic)

Ilustración 17 Comandos en iOS y Android

Lamentablemente no es recomendable utilizar el emulador de Android debido que es demasiado lento para el desarrollo, ventajosamente existen

Algunas excelentes alternativas que cumplen la misma función que el mencionado con anterioridad y estos si son rápidos. Es recomendado probar y desarrollar directamente en un dispositivo Android nuestra aplicación, sin olvidar que es necesario que este se encuentre conectado a la computadora.

Debido a que una Ionic app es una página web, necesitamos tener un `index.html` que es lo que se va a cargar cuando inicie nuestra aplicación, para lo esto debemos agregando el siguiente contenido: (Ionic)

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Todo</title>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">

    <script src="lib/ionic/js/ionic.bundle.js"></script>

    <!-- Needed for Cordova/PhoneGap (will be a 404 during development) -->
    <script src="cordova.js"></script>
  </head>
  <body>
  </body>
</html>

```

Ilustración 18 HTML base ionic

Los archivos Córdoba son los últimos en ser agregados y se encuentran vacíos, estos se crean al momento que hacemos que la aplicación corra.

Para crear side menús en Ionic, se usa de directiva `<ion-side-menus>`. El código para crear este tipo de menú es sencillo. También se debe agregar en el `index.html`, dentro del tag `<body>` el siguiente código: (Ionic)

```

<body>
  <ion-side-menus>
    <ion-side-menu-content>
    </ion-side-menu-content>
    <ion-side-menu side="left">
    </ion-side-menu>
  </ion-side-menus>
</body>

```

Ilustración 19 Menú ionic

Estos códigos son los encargados de mostrar el menú, manejar el evento. Además, la aplicación cuenta con una directiva que es la encargada de contenerla.

Mientras el proceso de desarrollo avanza tenemos 4 formas de probar la aplicación:

- A través de un navegador.

- El navegador de un dispositivo móvil.
- Por medio de un emulador.
- Como una nativa aplicación en el celular.

Angular Js necesita de dos recursos de la web para crear un código contendor central y resultar de esta manera:

```
<!-- Center content -->
<ion-side-menu-content>
<ion-header-bar class="bar-dark">
<h1 class="title">Todo</h1>
</ion-header-bar>
<ion-content>
<!-- our list and list items -->
<ion-list>
<ion-item ng-repeat="task in tasks">
  {{task.title}}
</ion-item>
</ion-list>
</ion-content>
</ion-side-menu-content>
```

Ilustración 20 Codigos angularJs

A esto se le debe agregar una tarea para que realice nuestra aplicación, por lo que es necesario crear un controlador de angular, asimismo, podemos añadir algunas tareas al controlador en el archivo para realizar pruebas:

```
angular.module('todo', ['ionic'])

.controller('TodoCtrl', function($scope) {
  $scope.tasks = [
    { title: 'Collect coins' },
    { title: 'Eat mushrooms' },
    { title: 'Get high enough to grab the flag' },
    { title: 'Find the Princess' }
  ];
});
```

Ilustración 21 Tareas asignadas

Una ventana modal permite realizar un pequeño formulario el cual se lo usa para agregar nuevas tareas.

```
<script id="new-task.html" type="text/ng-template">

  <div class="modal">

    <!-- Modal header bar -->
    <ion-header-bar class="bar-secondary">
      <h1 class="title">New Task</h1>
      <button class="button button-clear button-positive" ng-click="closeNewT.
    </ion-header-bar>

    <!-- Modal content area -->
    <ion-content>

      <form ng-submit="createTask(task)">
        <div class="list">
          <label class="item item-input">
            <input type="text" placeholder="What do you need to do?" ng-model:
          </label>
        </div>
        <div class="padding">
          <button type="submit" class="button button-block button-positive">C
        </div>
      </form>

    </ion-content>

  </div>

</script>
```

Ilustración 22 Venta modal

El formulario se encarga de almacenar los valores de las tareas. En el contenedor principal se agrega un botón para así conseguir activar el modal., quedaría de esta manera:

```

<!-- Center content -->
<ion-side-menu-content>
  <ion-header-bar class="bar-dark">
    <h1 class="title">Todo</h1>
    <!-- New Task button-->
    <button class="button button-icon" ng-click="newTask()">
      <i class="icon ion-compose"></i>
    </button>
  </ion-header-bar>
  <ion-content>
    <!-- our list and list items -->
    <ion-list>
      <ion-item ng-repeat="task in tasks">
        {{task.title}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-side-menu-content>

```

Ilustración 23 Codigos formularios

El controlador queda así: (Ionic)

```

angular.module('todo', ['ionic'])

.controller('TodoCtrl', function($scope, $ionicModal) {
  // No need for testing data anymore
  $scope.tasks = [];

  // Create and load the Modal
  $ionicModal.fromTemplateUrl('new-task.html', function(modal) {
    $scope.taskModal = modal;
  }, {
    scope: $scope,
    animation: 'slide-in-up'
  });

  // Called when the form is submitted
  $scope.createTask = function(task) {
    $scope.tasks.push({
      title: task.title
    });
    $scope.taskModal.hide();
    task.title = "";
  };

  // Open our new task modal
  $scope.newTask = function() {
    $scope.taskModal.show();
  };

  // Close the new task modal
  $scope.closeNewTask = function() {
    $scope.taskModal.hide();
  };
});

```

Ilustración 24 Codigos controlador

Utilizando códigos parecidos a los generados en los campos anteriores, es decir, en la selección y creación de tareas, tendremos como resultado las funciones de añadir y elegir de proyectos y asimismo obtendremos un botón que cree nuevos proyectos. De esta manera debe quedar:

```

<!-- Center content -->
<ion-side-menu-content>
  <ion-header-bar class="bar-dark">
    <button class="button button-icon" ng-click="toggleProjects()">
      <i class="icon ion-navicon"></i>
    </button>
    <h1 class="title">{{activeProject.title}}</h1>
    <!-- New Task button -->
    <button class="button button-icon" ng-click="newTask()">
      <i class="icon ion-compose"></i>
    </button>
  </ion-header-bar>
  <ion-content scroll="false">
    <ion-list>
      <ion-item ng-repeat="task in activeProject.tasks">
        {{task.title}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-side-menu-content>

```

Ilustración 25 Contenedor central

```

<!-- Left menu -->
<ion-side-menu side="left">
  <ion-header-bar class="bar-dark">
    <h1 class="title">Projects</h1>
    <button class="button button-icon ion-plus" ng-click="newProject()">
    </button>
  </ion-header-bar>
  <ion-content scroll="false">
    <ion-list>
      <ion-item ng-repeat="project in projects" ng-click="selectProject(pro
        {{project.title}}
      </ion-item>
    </ion-list>
  </ion-content>
</ion-side-menu>

```

Ilustración 26 Side menú

Los códigos en el contenedor nos permiten añadir, almacenar y cargar los proyectos:

Esta es la culminación de la aplicación ya está funcional y pulida, solo resta ejecutar los códigos

- *Ionic en prototipos*

Ionic se presenta como una herramienta interesante para ejecutar esta metodología, ya que al ser un marco de desarrollo de aplicaciones híbridas permite la realización de prototipos que se pueden probar en diferentes sistemas operativos.

Ionic posee una herramienta muy útil para poder tener prototipos corriendo en ambientes reales dentro de su marco de herramientas, Ionic posee una aplicación denominada Ionic App la cual se descarga en cualquier dispositivo móvil que ejecute Android o IOS, esta aplicación tiene la particularidad de poder cargar proyectos que previamente hayan sido subidos desde la consola de desarrollo a la nube de Ionic, con un simple código la aplicación descarga el proyecto, lo empaqueta y lo ejecuta dentro de esta aplicación simulando una ejecución real de la aplicación.

Ionic App ayuda que los prototipos puedan ser probados de una manera más rápida y ejecutándose en un ambiente real. Esto minoriza el riesgo a la hora de entrega de la aplicación, teniendo un canal mucho más cercano a que el cliente valide sus necesidades y mantenga una constante actualización de los cambios y trabajos nuevo que se realizan sobre la aplicación. Estos prototipos son un poco más avanzados ya que ejecutan funcionalidad directamente dentro de Ionic App, estos prototipos son generados especialmente por desarrolladores, pero Ionic también posee una herramienta

orientada a diseñadores que permite realizar prototipos netamente de diseño de interfaz, esta herramienta es Ionic Creator.

Ionic Creator tiene la particularidad de que también puede integrarse con Ionic App y permitir ejecutar el diseño realizado en un entorno más real, esta herramienta puede generar también proyectos base que integran los estilos e interfaces diseñadas en ella lo que es de gran ayuda para los desarrolladores y representa un gran apoyo a la hora de hacer prototipos más funcionales en cuanto a tiempo de desarrollo y pruebas.

Ionic tiene una gran ventaja en la generación de prototipos por lo que usar esta metodología la transforma en una herramienta ideal a lo largo del desarrollo sobre todo porque los tiempos son relativamente cortos.

En estos últimos años se ha tornado frecuente encontrar aplicaciones desarrolladas por medio de la implementación del modelo de prototipos, por lo que es indispensable el manejo de este tipo de herramienta a la hora de programar, ya que esta técnica se la puede efectuar en cualquier momento del desarrollo, debido a la facilidad que tiene esta de englobar cada vez más las particularidades del producto final. Asimismo, generan un alto nivel de funcionalidad, obteniendo beneficios como la satisfacción del usuario con respecto a sus requerimientos debido a su reusabilidad, flexibilidad y adaptabilidad.

Por lo tanto, podemos decir que este prototipo es estrechamente similar al producto terminado, pero se debe seguir experimentando y reformando para hasta realizar el siguiente

prototipo y de esta forma conseguir a uno que sea considerado como el software final.

2.6.3. Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Se permite el uso de tecnologías web estándar - HTML5, CSS3 y JavaScript para el desarrollo multiplataforma. Las aplicaciones se ejecutan dentro de envoltorios específicos para cada plataforma, y se basan en los enlaces de la API compatibles con los estándares de acceso a las capacidades de cada dispositivo, tales como sensores, datos, estado de la red, etc.

(Cordova, Apache Cordova)

Usa Córdoba si es:

- Un desarrollador móvil y quieren extender una aplicación a través de más de una plataforma, Sin tener que volver a ponerlo en práctica con el conjunto de herramienta de lenguaje y de cada plataforma. (Cordova, Apache Cordova)
- Un desarrollador web y desean implementar una aplicación web que está acondicionado para la venta en diferentes portales de la tienda de aplicaciones. (Cordova, Apache Cordova)
- Un desarrollador móvil interesados en mezclar los componentes de aplicaciones nativas con un WebView (ventana del navegador especial) que se puede acceder a las API a nivel de dispositivo, o si se quiere desarrollar una interfaz de complementos entre los componentes nativos y WebView. (Cordova, Apache Cordova)

En el siguiente diagrama podemos ver que Córdoba cuenta con varios componentes para una aplicación y el alto nivel de arquitectura de aplicaciones que posee.

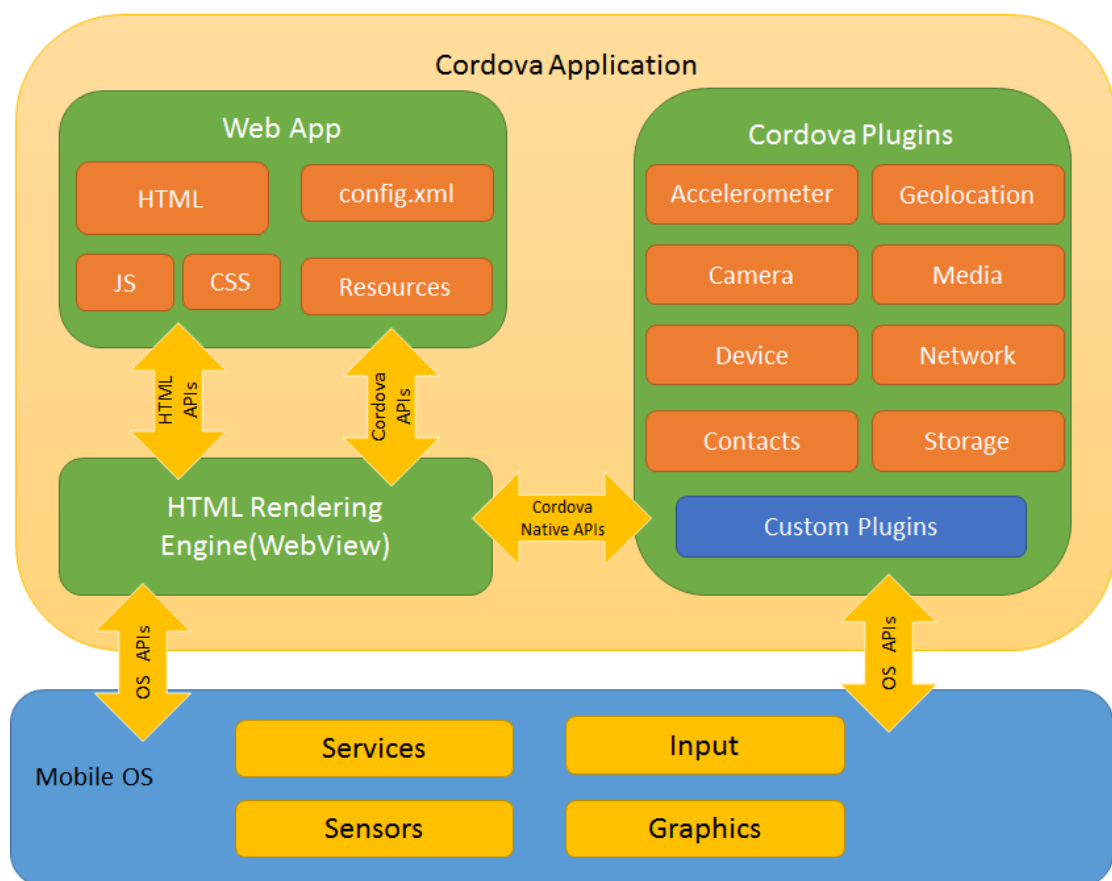


Ilustración 27 Arquitectura cordova

3. Diseño, desarrollo de la aplicación

3.1. Solicitud y condiciones del proyecto

El departamento de investigación de Sertecpet S.A requiere la creación de un aplicativo móvil denominado “Handbook” el cual tiene como objetivo ser una herramienta para el personal interno de la empresa, en la cual los ingenieros tendrán la facilidad de realizar cálculos y consultas que usualmente las realizan en su trabajo diario.

Para el desarrollo del proyecto el departamento designó cuatro recursos, estos recursos serán responsables del desarrollo del proyecto e interactuarán en todas sus fases para garantizar su calidad. Se asignó un total de 160 horas para el desarrollo del proyecto.

Los requerimientos y características de la aplicación deben estar adecuadas al tiempo otorgado por el departamento, para el cumplimiento de los objetivos planteados por el equipo se decidió el uso de metodologías ágiles durante el proceso de desarrollo.

3.2. Arquitectura propuesta para el desarrollo

3.3. Designación de roles

Para la ejecución del proyecto se designaron a los siguientes roles a cada uno de los integrantes del equipo:

ROL	PERSONA ASIGNADA
Líder de proyecto	Leonardo Mena
Analista Funcional	Jorge Soria
Pruebas y revisión	Annabel Arauz

Desarrollador	Jean Gonzalez
---------------	---------------

3.4. Requerimientos

El departamento de investigación y desarrollo a tras de reuniones previas concluyó que la aplicación móvil constará de cuatro módulos específicos, los cuales encajan dentro del tiempo asignado al proyecto, a continuación, se detallan con sus características y responsables.

3.5. Descripción requerimientos:

- *Transformación de Unidades*

Se necesita tener una herramienta para transformación debido a que los parámetros como presión, temperatura, longitud pueden ser medidos en varias unidades y acorde al sistema de unidades con el que se esté trabajando se requerirá transformarlas.

- *Cálculo de volúmenes.*

En campo se puede tener varios tipos de recipientes, como son: tanques, separadores, tubería; ésta herramienta permite que se pueda determinar la cantidad de fluido que se necesita para llenar el recipiente, su capacidad total o la cantidad de fluido que contiene el mismo.

- *Base de datos de tubería.*

En los pozos de petróleo se tienen dos tipos de tubos (casing y tubing), los casing son tubería de revestimiento que se coloca dentro del pozo mientras se está perforando, los tubing se utiliza para producir el crudo. Las dos clases de tubería se caracterizan acorde a lo siguiente: grado, peso, diámetros internos y externos, etc. Dichas propiedades se seleccionan de acuerdo al tipo de pozo y

profundidades que se vayan a manejar, con esto ya se saben todas las dimensiones (espesor de tubería), peso de toda la sarta, herramientas que pueden ingresar.

- *Cálculo IPR.*

Sirve para graficar la curva IPR y determinar la cantidad máxima de petróleo que se va a producir diariamente de ese pozo.

Línea recta se utiliza cuando el fluido del pozo no contiene gas en solución

Curva se utiliza cuando el fluido tiene gas en solución, obteniéndose así un P_b presión de burbuja (presión a la cual el gas empieza separarse del fluido)

3.6. Tabla de requerimientos

ID	PRIORIDAD	MODULO	DESCRIPCION	RESPONSABLE
1	Alta	Convertidor de unidades	<p>Este módulo tiene el objetivo de proporcionar a los ingenieros la posibilidad de realizar transformaciones de unidades de los siguientes tipos:</p> <ul style="list-style-type: none"> • Presión • Volumen • Temperatura • Fluido • Longitud 	Jean Gonzalez
2	Alta	Calculo de capacidades	<p>El principal objetivo de este módulo es permitir el cálculo del volumen y capacidad de los siguientes elementos:</p> <ul style="list-style-type: none"> • Cilindro vertical • Cilindro horizontal • Cilindro horizontal con hemisferios 	Jean Gonzalez

			<ul style="list-style-type: none"> • Cilindro vertical con hemisferios • Pipeline • Sección anular 	
3	Media	Consulta de características de tuberías Casing y Tubing	<p>Este módulo sirve a los ingenieros como una fuente consulta de características de tuberías casing y tubing, podrán consultar en los tipos API y Tamsa, los atributos a conocer serán:</p> <ul style="list-style-type: none"> • Nominal del diámetro interno • Diámetro del drift • Resistencia al colapso • Rendimiento de presión interna • Volumen 	Jean Gonzalez
4	Alta	Cálculos IPR	<p>Este módulo tiene como objetivo el cálculo curvas IPR, la aplicación permitirá el cálculo de estas a través de los siguientes modelos.</p> <ul style="list-style-type: none"> • Lineal • Vogel • Vogel compuesto 	Jean Gonzalez

3.7. Planificación

Para el desarrollo de la aplicación el departamento de investigación y desarrollo decidió el total del desarrollo en cuatro iteraciones las cuales se ejecutarán siguiendo la arquitectura seleccionada anteriormente.

A continuación, se detallan las tareas a ejecutar en cada iteración, así como el tiempo asignado a cada tarea.

3.7.1. Iteración N°1

Iteración	Tareas	Nro. Horas	Documentos/Entregables
1	Reunión Inicial de iteración	1	Acta de Reunión
1	Diseño de interfaz (Convertidor de unidades)	6	Esquema de diseño
1	Desarrollo modulo: (Convertidor de unidades)	24	Prototipo funcional
1	Pruebas	8	Informe de Pruebas
1	Reunión final de iteración	1	Acta de Reunión

3.7.2. Iteración N°2

Iteración	Tareas	Nro. Horas	Documentos/Entregables
2	Reunión Inicial de iteración	1	Acta de Reunión
2	Diseño de interfaz (Calculo de capacidades)	6	Esquema de diseño
2	Desarrollo modulo: (Calculo de capacidades)	30	Prototipo funcional
2	Pruebas	10	Informe de Pruebas
2	Reunión final de iteración	1	Acta de Reunión

3.7.3. Iteración N°3

Iteración	Tareas	Nro. Horas	Documentos/Entregables
3	Reunión Inicial de iteración	1	Acta de Reunión
3	Diseño de interfaz (Consulta características de tubería)	6	Esquema de diseño
3	Desarrollo modulo: (Consulta características de tubería)	15	Prototipo funcional
3	Pruebas	6	Informe de Pruebas
3	Reunión final de iteración	1	Acta de Reunión

3.7.4. Iteración N°4

Iteración	Tareas	Nro. Horas	Documentos/Entregables
------------------	---------------	-------------------	-------------------------------

3	Reunión Inicial de iteración	1	Acta de Reunión
3	Diseño de interfaz (Consulta características de tubería)	6	Esquema de diseño
3	Desarrollo modulo: (Consulta características de tubería)	25	Prototipo funcional
3	Pruebas	10	Informe de Pruebas
3	Reunión final de iteración	1	Acta de Reunión

3.8. Diseño Base

Para el diseño base del prototipo de definieron las siguientes maquetas:

3.8.1. Pantalla Inicial (Splash)

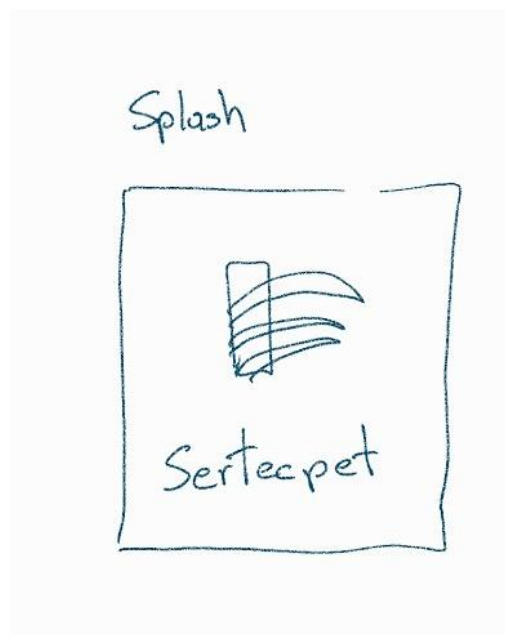


Ilustración 28 Pantalla Inicial (Splash)

Para la pantalla inicial se solicitó un logo representativo de la empresa con fondo blanco.

3.8.2. Esquema base de interacción

Para la interacción base de la aplicación el departamento definió que se realicen 4 pestañas que dieran acceso a cada módulo, y en la parte superior el nombre del módulo en el que se encontraba el usuario.

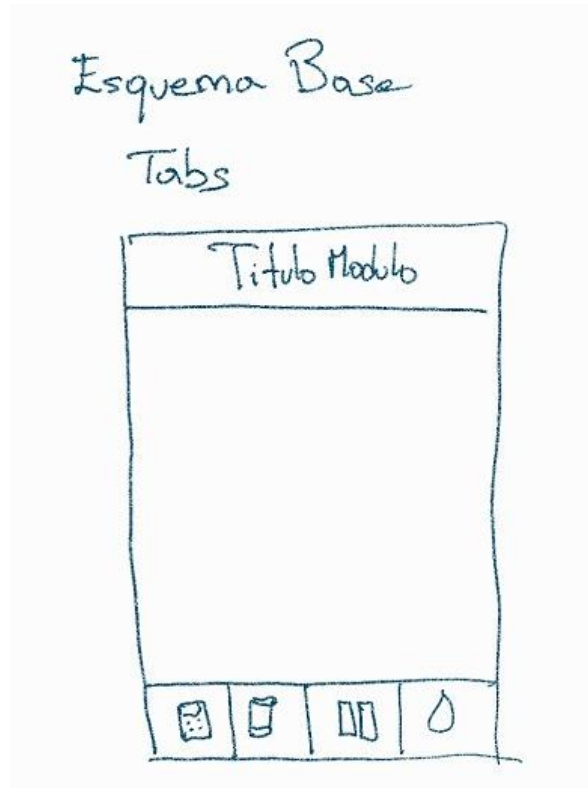


Ilustración 29 Esquema base interacción (Tabs9)

Cada pestaña contara con interacciones genéricas, cada pestaña tendrá una lista de opciones para los cálculos, cada vez que se ingrese dentro de una opción la aplicación habilitara el botón back que servirá para regresar al menú de opciones del módulo, también se habilitara el botón borrar el cual eliminara todos los datos ingresados, esto con la finalidad de realizar cálculos de una manera más rápida.

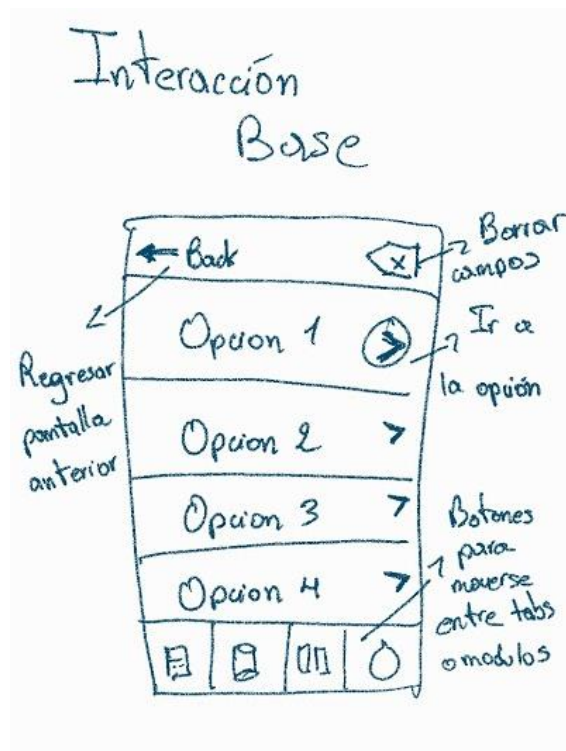


Ilustración 30 Esquema base de interacción (Botones y funcionalidad)

3.9. Desarrollo Base

Para el desarrollo base se creó una aplicación Ionic con diseño tabs, para esto Ionic construye un proyecto en cual crea una vista padre de tabs y permite ir agregando hijos. El proyecto Ionic cuenta con la siguiente estructura de archivos:

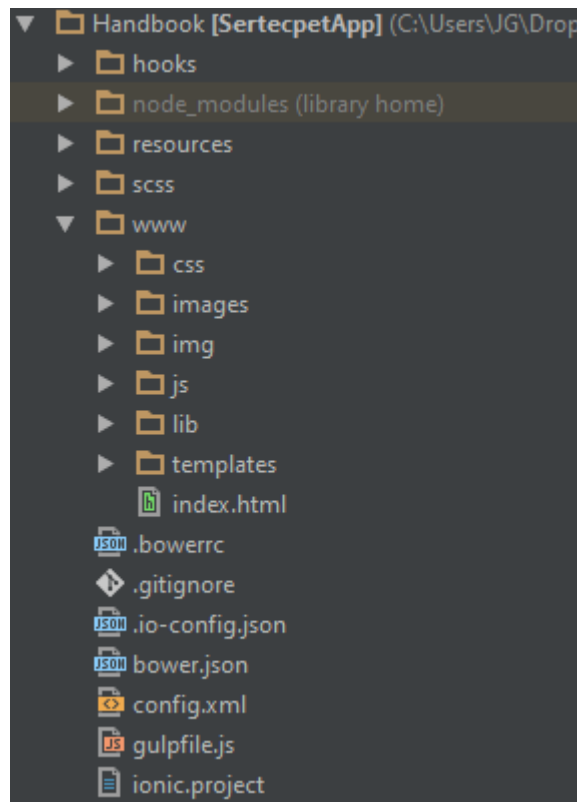


Ilustración 31 Estructura de archivos del proyecto

Dentro de las carpetas generadas por Ionic la carpeta en la cual tendremos que agregar contenido es la carpeta “www”, esta carpeta consta de las siguientes subcarpetas

- **css**: Dentro de esta carpeta se agregarán todos los estilos que requiera la aplicación.
- **images**: Esta carpeta contiene las imágenes, fotografías o archivos visuales que requiera la aplicación
- **js**: Esta carpeta contiene todos los archivos de tipo script que necesita la aplicación.
- **lib**: Esta carpeta almacena todas las librerías s utilizar dentro del proyecto.
- **templates**: Dentro de esta carpeta se encuentran todos los archivos HTML del proyecto aquí se crearán las vistas que irán dentro de cada pestaña de la aplicación.

Los archivos que contendrán toda la lógica de la aplicación se encuentran esta dirección “www/js” y son los siguientes:

- **app.js:** Dentro de este archivo se declara el modulo padre de Angular JS denominado “Sertecpet”, en esta sección se realizará la inyección de dependencias globales para todos los controladores. En este archivo también se realizará el enrutamiento a los distintos archivos HTML
- **controllers.js:** En este archivo se crearán todos los controladores necesarios para la aplicación Angular Js
- **filters.js:** Este archivo contiene todos los filtros que la aplicación requiera para dar formatos a los textos dentro de la aplicación Angular JS
- **services.js:** Este archivo almacenara todos los diccionarios de datos, formulas, catálogos, en general toda la información que requieran los controladores, si fuera necesario realizar el consumo de un web service se la realiza dentro de este archivo.

3.10. Iteraciones

Cumpliendo con la planificación establecida para el proyecto a continuación se detalla el desarrollo de cada iteración de este proyecto.

3.10.1. Iteración N°1

- *Reunión inicial de iteración*

En esta reunión se revisaron los requerimientos funcionales del módulo de transformación de unidades, se determinó que la transformación de unidades funcionara de la siguiente manera:

Cada categoría de transformación de unidades (presión, volumen, temperatura entre otros) contarán con una unidad determinada como base, todas las transformaciones se harán en función de esta unidad base, todas las unidades tendrán una fórmula de transformación de la unidad requerida a la unidad base denominada

“formula de vuelta” y otra que realiza el camino inverso al anterior que trasforma el valor base a la unidad requerida denominada formula de vuelta

Ejemplo:

ft es la unidad base de Longitudes

Si requerimos transformar

Pulgadas(in) a Centímetros(cm)

Se trasforman las pulgadas a ft utilizando la fórmula de vuelta

$$\text{Valor}/12 = \text{valor (ft)}$$

El resultado se remplaza en la fórmula de ida de los (cm)

$$\text{Valor} * 30.4800000012 = \text{Valor (cm)}$$

Para el desarrollo del módulo se establecen las siguientes tareas:

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F001	Configuración de proyecto base	Se toma el proyecto base y se lo personaliza de acuerdo a requerimientos del proyecto	Alta	8	0%	
F001	Creación vista de listado de categorías	E crea la pestaña para que se puedan enlistar las categorías de unidades	Alta	2	0%	
F001	Carga dinámica de categoría de unidades	Los listados de categorías de unidades no deben ser estáticos, el desarrollo de la aplicación debe permitir cargas las categorías de unidades que se necesite	Media	1	0%	
F002	Transformación simultanea de una unidad	Al momento de ingresar un valor en cualquiera de las unidades disponibles se deben calcular automáticamente los calores de todas las unidades disponibles	Alta	13	0%	
F003	Borrado de campos	Elimina automáticamente toda la información ingresada en la pestaña de unidades	Media	3		

- *Diseño de interfaz*

Para este módulo se esquematizo las siguientes pantallas:

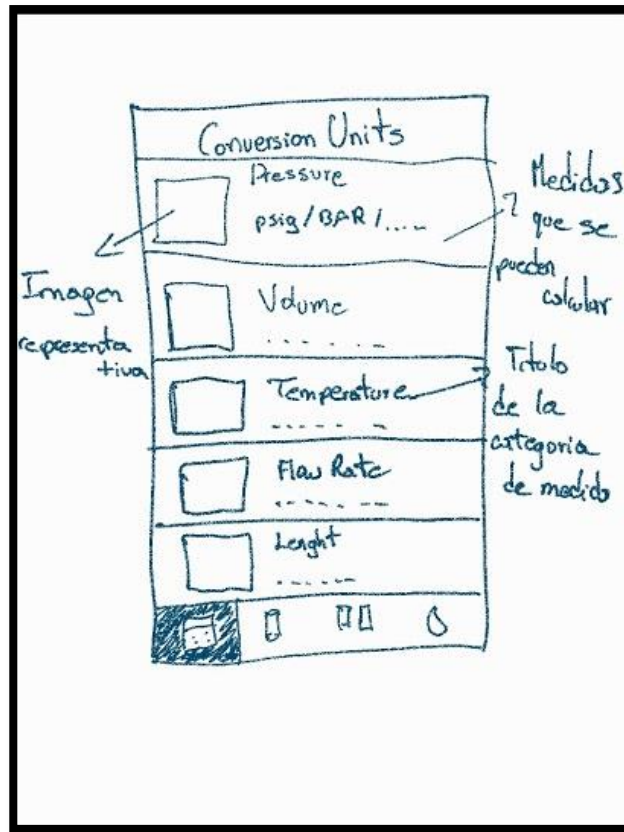


Ilustración 32 Diseño módulo conversión de unidades

Donde se espera una lista de las categorías a convertir, se plantea que cada elemento de la lista mantenga una imagen representativa, un título principal y una descripción en la que consten las unidades que pueden participar de la transformación.

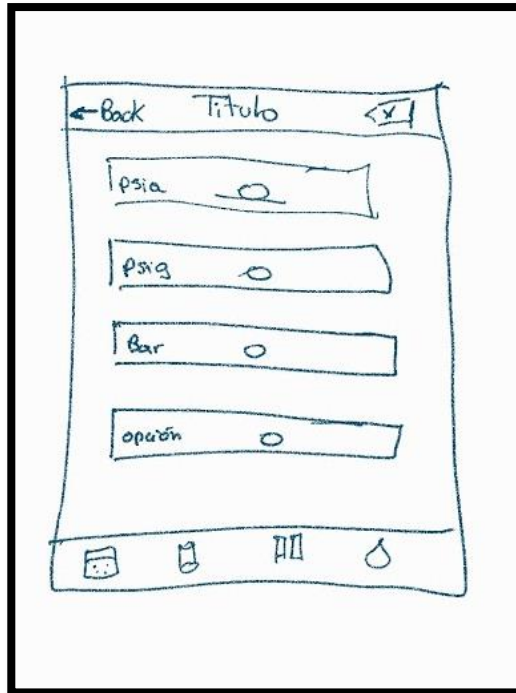


Ilustración 33 Diseño módulo conversión de unidades (vista unitaria)

Al momento de seleccionar una de las categorías se despliega una pantalla que contiene todas las unidades disponibles para participar de la transformación, el objetivo es que el usuario al ingresar información en un solo campo pueda tener la equivalencia en todas las unidades disponibles.

- *Desarrollo modulo*

Para la creación de este módulo se creó el archivo HTML “www/templates/tab-medidas.html”, este archivo es el que muestra la lista de categorías para la transformación de unidades.

```

<ion-view view-title="Conversion Units">
  <ion-content>
    <ion-list>
      <ion-item class="item-remove-animate item-icon-right item-thumbnail-left" ng-repeat="medida in
medidas" type="item-text-wrap" href="#/tab/units/{{medida.id}}">
        <img src={{medida.face}} style="width:80px; height:80px;">
        <h2>{{medida.name}}</h2>
        <p>{{medida.lastText}}</p>
        <i class="icon ion-chevron-right icon-accessory"></i>
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>

```

El archivo “www/templates/medidas-units.html” contiene el código HTML que despliega la aplicación al seleccionar una categoría

```
<ion-view view-title="{{medida.name}}">
  <ion-nav-buttons side="right"> <!-- right -->
    <button menu-toggle="left" class="button button-icon icon ion-backspace" ng-
click="borrar()"></button>
  </ion-nav-buttons>
  <ion-content class="padding">
    <ion-list>
      <ion-item class="item-remove-animate item-icon-right" ng-repeat="unidad in medida.unidades |
orderBy:'id'" type="item-text-wrap">
        <label class="item item-input">
          <span class="input-label">{{unidad.name}}</span>
          <input placeholder="0" ng-model="unidad.value" ng-change="transformar(unidad)"
type="number">
        </label>
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>
```

Dentro del archivo “www/js/app.js” se agregan las rutas de acceso a la vista “www/templates/tab-medidas.html”

```
.state('tab.medidas', {
  url: '/medidas',
  views: {
    'tab-units': {
      templateUrl: 'templates/tab-medidas.html',
      controller: 'MedidasCtrl'
    }
  }
})
```

Para la vista “www/templates/tab-medidas.html” se crea el siguiente controlador: “MedidasCtrl” el cual carga la lista de categorías dentro de la vista.

```
.controller('MedidasCtrl', function ($scope, Medidas) {
  $scope.medidas = Medidas.all();
  $scope.remove = function (medida) {
    Medidas.remove(medida);
  }
})
```

Al momento de seleccionar una categoría la vista “www/templates/medidas-units.html” utiliza el controlador “MedidaUnitCtrl” el cual recibe la categoría seleccionada y realiza la conversión de la unidad ingresada a todas las unidades disponibles.

```
.controller('MedidaUnitsCtrl', function ($scope, $stateParams, Medidas,$filter) {
  $scope.medida = Medidas.get($stateParams.medidaID);
  $scope.borrar= function(){

    for(var i= 0; i < $scope.medida.unidades.length; i++)
    {
      $scope.medida.unidades[i].value=null;
    }
  }
  $scope.transformar = function (unidadModificada) {
    var self2 = this;

    var value = this.$eval('unidad.value');

    var value = eval(this.medida.unidades[unidadModificada.id].formulaVuelta);

    for (index in this.medida.unidades) {
      var unidad = this.medida.unidades[index];

      if (unidad != unidadModificada && value != null) {
        unidad.value = $filter('setDecimal')(eval(unidad.formula),3);
      }
      if (unidad != unidadModificada && (value == null || value <= 0))
        unidad.value = 0;
    }
  }
})
```

En el archivo “www/js/services.js” se crea el Factory “Medidas” el cual almacena la información de todas las unidades disponibles, fórmulas de ida y vuelta, además contiene los métodos genéricos para las transformaciones de cualquier unidad de las distintas categorías.

```

var medidas = [
  {
    id: 0,
    name: 'Pressure',
    seleccionado: 0,
    lastText: 'psia / psig / BAR / kPa / (kg/cm2) / atm',

    face: 'images/PRESION.png',
    unidades: [

      {
        id: 0,
        name: 'psia',
        formula: 'value',
        value: null,
        formulaVuelta: 'value'
      },
      {
        id: 1,
        name: 'psig',
        formula: '(value)- 14.6959',
        value: null,
        formulaVuelta: '(value) + 14.6959'
      },
      {
        id: 2,
        name: 'bar',
        formula: '(value)*0.06894756999871',
        value: null,
        formulaVuelta: '(value)/0.06894756999871'
      },
      {
        id: 3,
        name: 'kPa',
        formula: '(value)*6.894756999871',
        value: null,
        formulaVuelta: '(value)/6.894756999871'
      },
      {
        id: 4,
        name: 'kg/cm2',
        formula: '(value)*0.07030695497464',
        value: null,
        formulaVuelta: '(value)/0.07030695497464'
      },
      {
        id: 5,
        name: 'atm',
        formula: '(value)*0.06804596101413',
        value: null,
        formulaVuelta: '(value)*0.06804596101413'
      }
    ]
  },
],

```

```

return {
  all: function () {
    return medidas;
  },
  remove: function (medidas) {

```

```

    chats.splice(medidas.indexOf(medidas), 1);
  },
  get: function (medidaID) {
    for (var i = 0; i < medidas.length; i++) {
      if (medidas[i].id === parseInt(medidaID)) {
        return medidas[i];
      }
    }
    return null;
  },
},

transformarBase:function(value,idUnidad,tipo){

  switch (tipo)
  {
    case "presion":{

      var result = eval(medidas[0].unidades[idUnidad].formulaVuelta);

      return result;

      break;
    }

    case "caudal":{

      var result = eval(medidas[3].unidades[idUnidad].formulaVuelta);

      return result;

      break;
    }

    case "volumen":{

      var result = eval(medidas[1].unidades[idUnidad].formulaVuelta);

      return result;

      break;
    }
    case "longitud":{

      var result = eval(medidas[4].unidades[idUnidad].formulaVuelta);

      return result;

      break;
    }
  }
},

transformarIda:function(value,idUnidadNew,idUnidadOld,tipo){

  switch (tipo)
  {

```

```

case "presion":{
    value=eval(medidas[0].unidades[idUnidadOld].formulaVuelta);

    var result = eval(medidas[0].unidades[idUnidadNew].formula);

    return result;

    break;
}

case "caudal":{

    value=eval(medidas[3].unidades[idUnidadOld].formulaVuelta);

    var result = eval(medidas[3].unidades[idUnidadNew].formula);

    return result;

    break;
}
case "longitud":{

    value=eval(medidas[4].unidades[idUnidadOld].formulaVuelta);

    var result = eval(medidas[4].unidades[idUnidadNew].formula);

    return result;

    break;
}
case "volumen":{
    value=eval(medidas[1].unidades[idUnidadOld].formulaVuelta);

    var result = eval(medidas[1].unidades[idUnidadNew].formula);

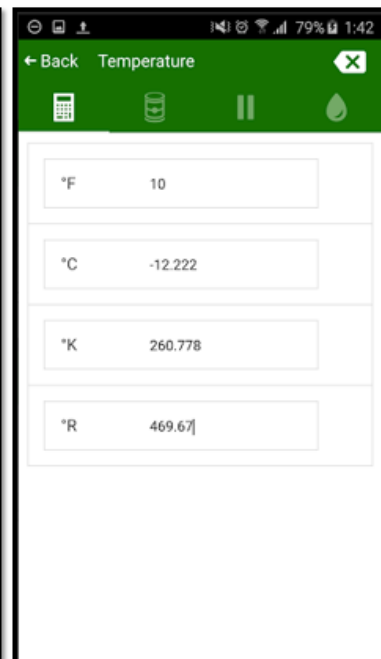
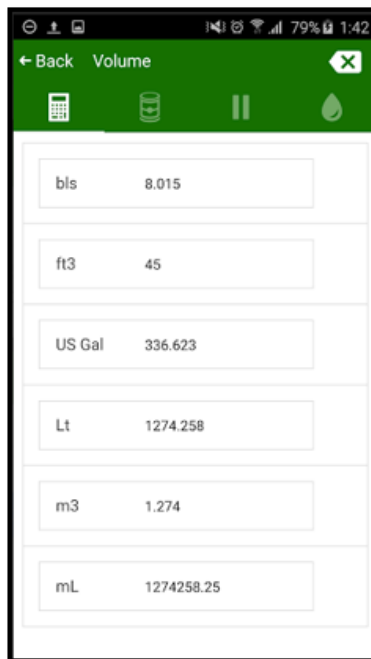
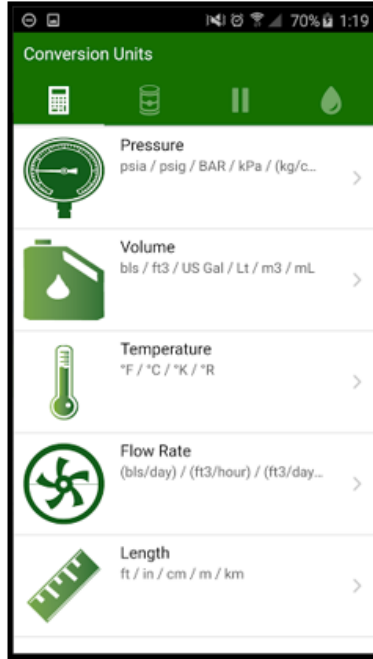
    return result;

    break;
}
}
};
})

```

Se entrega el primer prototipo de la aplicación con el módulo de transformación de unidades funcional.

T e s t	A c t i o n	R e s u l t a d o	R e s u l t a d o	P a s a l a
1	T r a n s f o r m a r r 1 0 0	P s i g: 8 5. 3 0 4 B a r : 6 .8 9 5	P s i g: 8 5. 3 0 4 B a r : 6 .8 9 5	S i



	p s i a	k P a: 6 8 9. 4 7 6 k g/ c m 2: 7. 0 3 1 at m :6 .8 0 5	k P a: 6 8 9. 4 7 6 k g/ c m 2: 7. 0 3 1 at m :6 .8 0 5	
2	T r a n s f o r m a r 4 5 (b l s)	F t3 : 2 5 2. 6 5 6 U s G al : 1 8 9 0 L t: 7 1	F t3 : 2 5 2. 6 5 6 U s G al : 1 8 9 0 L t: 7 1	S i

		5 4. 4 2 8 M 3: 7. 1 5 4 M l: 7 1 5 4 4 2 8. 5	5 4. 4 2 8 M 3: 7. 1 5 4 M l: 7 1 5 4 4 2 8. 5	
3	T r a n s f o r m a r 4 0 ° C	° F : 1 0 4 ° k: 3 1 3 ° R : 5 6 3. 6 7	° F : 1 0 4 ° k: 3 1 3 ° R : 5 6 3. 6 7	S i
4	T r a n s f	F t3 /h o u r:	F t3 /h o u r:	N o (e l

	o r m a r 5 6 (m 3 / d a y)	1 9 7 7. 4 9 7 F t3 /d a y: 4 7 4 5 9. 2 6 M 3/ h o u r: 1 3 4 3. 8 9 U s G al / m in : 2 4 6. 5 6 9	1 9 7 7. 4 9 7 F t3 /d a y: 4 7 4 5 9. 2 6 M 3/ h o u r: 1 3 4 3. 8 9 U s G al / m in : 2 4 6. 5 6 9	c á l c u l o d e (m 3 / h o u r) n o c o i n c i d e)
5	T r	F t:	F t:	N o

a	8	8	(
n	C	C	E
s	m	m	l
f	:2	:2	r
o	4	4	e
r	3.	3.	s
m	8	8	u
a	4	4	l
r	M	M	t
9	:	:	a
6	2.	2	d
(4	4	o
i	3	3.	e
n	8	8	n
)	K	4	m
	m	K	e
	:	m	t
	0.	:	r
	0	0.	o
	0	0	s
	2	0	s
		2	e
			r
			r
			ó
			n
			e
			o
)

Se receptaron las pruebas y se corrigieron los errores rápidamente al detectar que el problema se encontraba en un mal ingreso de la formula.

- *Reunión final de iteración*

Durante esta reunión el área de investigación y desarrollo decidió continuar con el diseño planteado para los siguientes módulos, el área de pruebas presento el informe de pruebas y como

resultado se aprobó el primer prototipo de la aplicación, como feedback se receptaron las siguientes recomendaciones:

- Implementar el sistema de unidades en todo lo módulos: El departamento decidió que era clave para el proyecto que los demás módulos trabajen con unidades dinámicas para los cálculos requeridos

3.10.2. Iteración N°2

- *Reunión inicial de iteración*

Para el inicio de la segunda iteración el quipo realiza la revisión de los requerimientos para el desarrollo de este módulo, el área matemática ha desarrollado las formulas matemática y las entregó en JavaScript para ser usadas dentro de la aplicación.

A continuación, el detalle de las formulas:

Cilindro vertical:

La aplicación solicita como datos de ingreso diámetro, altura y longitud

```
d=diámetro:  
h=altura:  
l= longitud:  
  
volumen= Math.PI * Math.pow(d,2)*h)/(4*(5.61458284))  
  
capacidad= h*100
```

Cilindro horizontal

La aplicación solicita como datos de ingreso diámetro, altura y longitud.

```
var a;  
var y;  
var angulo;  
var aocupada;  
var aseleccion;  
var vtotal;
```

```

var vtotalOcup

if(h>(d/2))
a=(h-(d/2));
else
a=((d/2)-h);

y= 2* Math.sqrt((Math.pow(d,2)/4)-(Math.pow(a,2)));

angulo= (Math.asin(y/d)*2*180)/Math.PI;

aocupada=((Math.PI*Math.pow(d,2))/4)-(Math.pow(d,2)/8)*(Math.PI *angulo/180-
Math.sin((angulo*Math.PI)/180));
aseleccion =(Math.pow(d,2)/8)*(Math.PI*angulo/180-Math.sin((angulo*Math.PI)/180));
vtotal= Math.PI*Math.pow(d,2)*l/4;

if((d/2)<h)
vtotalOcup= aocupada*l;
else
vtotalOcup=aseleccion*l;

volumen= vtotalOcup/5.61458284;

capacidad=vtotalOcup*100)/vtotal;

```

Cilindro vertical con hemisferios

La aplicación solicita como datos de ingreso diámetro, altura, longitud y longitud de la sección circular.

```

function VolTanEliHor( D, L, r, y){
var volumen;

if( y > D)
{
y = D;
}

var R = D / 2;

if( y == 0){
volumen = 0;
}
else{
volumen = (3*Math.PI*L*Math.pow(R,3) + 3*Math.PI*R*r*Math.pow(y,2) -
Math.PI*r*Math.pow(y,3) -6*L*Math.pow(R,3)*Math.atan(Math.sqrt(2*R-
y)/Math.sqrt(y)) - 3*Math.sqrt(2*R-y)*(L*Math.pow(R,2) - L*R*y)*Math.sqrt(y))/(3*R);
}
}

```

```

    }

    return volumen
}

var vol_tot = VolTanEliHor(d,l,b,d)
var vol_par_hor = VolTanEliHor(d,l,b,h)
if (vol_tot != 0)
capacidad =(vol_par_hor / vol_tot * 100,3);
else
capacidad=0;

volumen= vol_par_hor

```

Cilindro horizontal con hemisferios

La aplicación solicita como datos de ingreso diámetro, altura, longitud y longitud de la sección circular.

```

function VolTanEliVer(D,L,r,y){

var volumen;
var alt_tot = 2*r + L;
if (y > alt_tot)
    y = alt_tot;

var R = D/2;

if(r == 0) {
    volumen = Math.PI * Math.pow(R,2) * y;
    return Medidas.transformarIda(volumen,0,1,'volumen');
}

var vol_esf = Math.PI * Math.pow(R,2) * Math.pow(r,2) * (3*r - r) / (3 * Math.pow(r,2));
if (y <= r)
    volumen = Math.PI * Math.pow(R,2) * Math.pow(y,2) * (3*r - y) / (3 * Math.pow(r,2));
else if (r < y && y <= r + L) {
    vol_cil = Math.PI * Math.pow(R, 2) * (y - r);
    volumen = vol_esf + vol_cil;
}
else if (r + L < y && y <= alt_tot) {
    pro_top = 2 * r + L - y;
    vol_par = Math.PI * Math.pow(R, 2) * Math.pow(pro_top, 2) * (3 * r - pro_top) / (3 *
Math.pow(r,2));
    vol_par_esf = vol_esf - vol_par;
    volumen = vol_esf + Math.PI * Math.pow(R, 2) * L + vol_par_esf;
}
}

```

```

    return volumen
}

var vol_tot = VolTanEliVer(d, l, b, (2*b + l))
var vol_par_ver = VolTanEliVer(d, l, b, h)
if (vol_tot != 0)
    capacidad = vol_par_ver / vol_tot * 100;
else
    capacidad=0;

volumen= vol_par_ver;

```

Pipeline

La aplicación solicita como datos de ingreso diámetro, altura, longitud y el diámetro interno del tubo.

```

volumen= Math.PI * Math.pow(id,2)*h) / (144*4*(5.61458284))
capacidad=(h*100) / (l);

```

Sección anular

La aplicación solicita como datos de ingreso el diámetro interno del “Casing” y el diámetro externo del “Tubing”

```

volumenAnular= ((Math.PI * (Math.pow(idCasing,2) -
Math.pow(odTubing,2)) * l) / (144*4*(5.61458284))

```

Para el desarrollo del módulo se establecen las siguientes tareas:

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones

F006	Filtro que redondea un valor en 3 decimales	Este filtro tiene la finalidad de homologar el número de decimales para presentar todos los resultados.	Alta	2	0%	
F007	Transformación unidades para variables de ingreso	Esta funcionalidad permite realizar cálculos en diferentes unidades	Muy Alta	6	0%	
F008	Creación de vistas de capacidades	Creación de la vista de menú de capacidades y vista de una capacidad específica.	Alta	8	0%	
F0010	Cálculos de capacidades	Utilizar las formulas del área matemática para realizar los cálculos y posteriormente presentar en pantalla.	Alta	14	0%	

- *Diseño de interfaz*

Para el diseño de este módulo se toma como referencia el anterior generando una lista de los tipos de capacidades donde cada ítem de la lista consta de una imagen que representa la capacidad el nombre de la capacidad y una pequeña descripción de la misma.

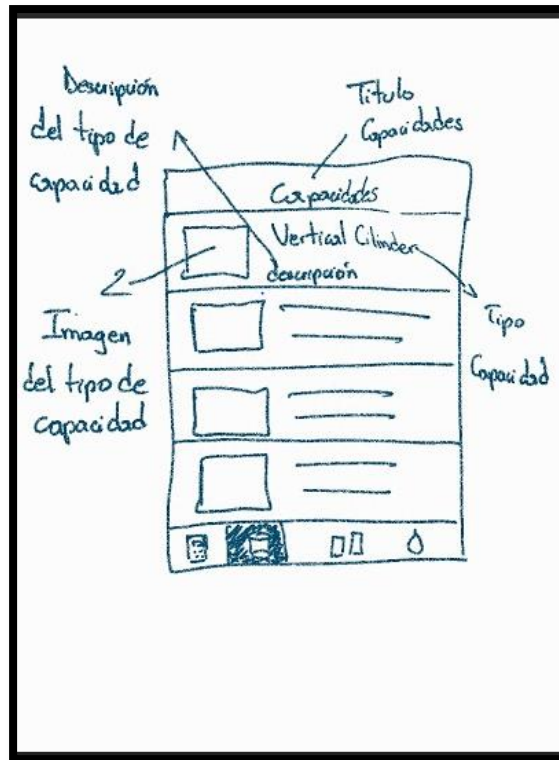


Ilustración 34 Diseño calculo volumen de capacidades (menú de cálculos)

Cada opción seleccionada mostrara campos de ingreso de datos de acuerdo a la necesidad el cálculo, para facilidad de los usuarios pueden ingresar valores en cualquier unidad de medida disponible, así como ver el resultado expresado en diferentes unidades.

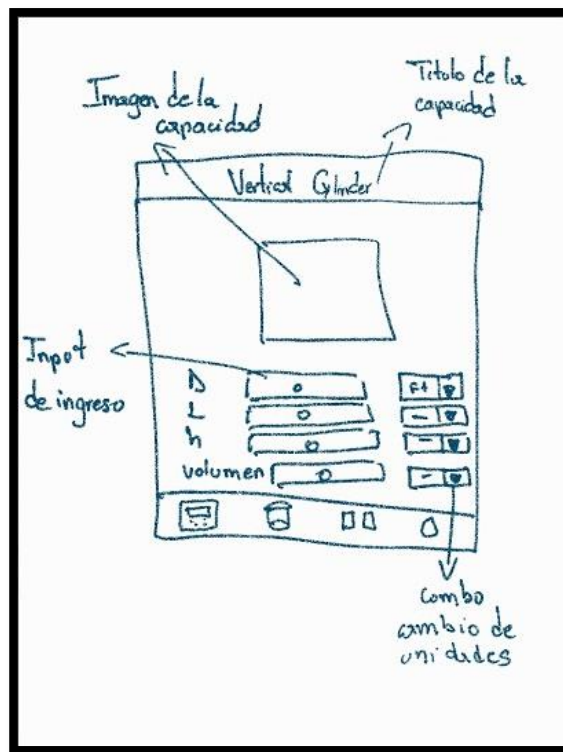


Ilustración 35 Diseño calculo volumen de capacidades (calculo especifico)

- *Desarrollo modulo*

Para que las presentaciones de los resultados mantengan un numero de decimales en toda la aplicación se realizó la creación de un filtro en Angular Js denominado “SetDecimal”, este archivo este filtro se encuentra en el archivo “www/js/filters.js”.

```
angular.module('sertecept.filters', [])

.filter('setDecimal', function ($filter) {
  return function (input, places) {
    if (isNaN(input)) return input;
    var factor = "1" + Array(+ (places > 0 && places + 1)).join("0");
    return Math.round(input * factor) / factor;
  };
});
```

El módulo de cálculo de volúmenes de capacidades cuenta de dos vistas, la primera muestra el listado de cálculos disponibles y se encuentra en el path “www/templates/tab-

capacidades.html”, esta vista toma las capacidades desde el controlador y las carga.

```

<ion-view view-title="Capacidades">
  <ion-content>
    <a class="item item-complex item-thumbnail-left" ng-repeat="capacidad in
    capacidades" href="#/tab/capacidadescalculos/{{capacidad.id}}">
      <div class="item-content">
        <img src={{capacidad.image}} style="background:#ffffff; width:80px;
        height:80px;">
        <h2>{{capacidad.name}}</h2>
        <p>{{capacidad.lastText}}</p>
      </div>
    </a>
  </ion-content>
</ion-view>

```

Al seleccionar una opción de la lista, la aplicación redirige al usuario a la página del cálculo específico, esta vista es la misma para todos los cálculos, dependiendo el cálculo seleccionado muestra los campos requeridos para el ingreso de datos, esta vista se encuentra en el siguiente path “”

```

<ion-view view-title="{{capacidad.name}}">
  <ion-nav-buttons side="right"> <!-- right -->
    <button menu-toggle="left" class="button button-icon ion ion-backspace" ng-
    click="borrar()"></button>
  </ion-nav-buttons>
  <ion-content class="padding">
    <img class="imagenTanque" src={{capacidad.image}} style="background:#ffffff;
    width:200px; height:200px;margin:auto; display: block">

    <div class="list">
      <div ng-show="capacidad.id==4" class="item item-input item-select">
        <span class="input-label">ID</span>
        <input ng-model="capacidad.variables.id" type="number" ng-
        change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)" >
        <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as
        unidad.name for unidad in medidas[4].unidades " ></select>
      </div>
      <div ng-show="capacidad.id==5" class="item item-input item-select">
        <span class="input-label">ID Casing</span>
        <input ng-model="capacidad.variables.idCasing" type="number" ng-
        change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)" >
        <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as
        unidad.name for unidad in medidas[4].unidades " ></select>
      </div>
      <div ng-show="capacidad.id==5" class="item item-input item-select">
        <span class="input-label">OD Tubing</span>
        <input ng-model="capacidad.variables.odTubing" type="number" ng-
        change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)" >
        <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as
        unidad.name for unidad in medidas[4].unidades " ></select>
      </div>
    </div>
  </ion-content>

```

```

    <div ng-show="capacidad.id!=4 && capacidad.id !=5" class="item item-input item-select">
      <span class="input-label">D</span>
      <input ng-model="capacidad.variables.d" type="number" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)">
      <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as unidad.name for unidad in medidas[4].unidades " ></select>

    </div>
    <div class="item item-input item-select">
      <span class="input-label">L</span>
      <input ng-model="capacidad.variables.l" type="number" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)">
      <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as unidad.name for unidad in medidas[4].unidades " ></select>
    </div>
    <div ng-show="capacidad.id==5" class="item item-input item-select">
      <span class="input-label">Volumen Anular</span>
      <input ng-model="capacidad.variables.volumenAnular" type="number" disabled="true" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)">
      <select ng-model="medidas[1].seleccionado" ng-options="unidad.id as unidad.name for unidad in medidas[1].unidades " ></select>

    </div>
    <div ng-show="capacidad.id == 2 || capacidad.id==3" class="item item-input item-select">
      <span class="input-label">B</span>
      <input ng-model="capacidad.variables.b" type="number">
      <select>
        <option selected>ft</option>
        <option>in</option>
        <option>om</option>
        <option>m</option>
      </select>
    </div>
    <div ng-show="capacidad.id!=5" class="item item-input item-select">
      <span class="input-label">h</span>
      <input ng-model="capacidad.variables.h" type="number" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)">
      <select ng-model="medidas[4].seleccionado" ng-options="unidad.id as unidad.name for unidad in medidas[4].unidades " ></select>
    </div>
    <div ng-show="capacidad.id!=5" class="item item-input item-select">
      <span class="input-label">Volumen</span>
      <input ng-model="capacidad.variables.volumen" type="number" disabled="true" ng-change="calcular(capacidad.id,medidas[4].seleccionado,medidas[1].seleccionado)" >
      <select ng-model="medidas[1].seleccionado" ng-options="unidad.id as unidad.name for unidad in medidas[1].unidades " ></select>
    </div>
    <div ng-show="capacidad.id!=5" class="item item-input item-select">
      <span class="input-label">Capacidad</span>
      <input ng-model="capacidad.variables.capacidad" type="number" disabled="true" >
      <select>
        <option selected>%</option>
      </select>
    </div>
  </div>
</ion-content>
</ion-view>

```

Este módulo para cargar la lista de cálculos de volumen de capacidades implementa el controlador “CapacidadesCtrl”, este controlador carga el catálogo de cálculos en la vista. El catalogo lo toma del servicio “Capacidades”

```
.controller('CapacidadesCtrl', function ($scope, Capacidades) {  
  
    $scope.capacidades = Capacidades.all();  
  
    $scope.selected = { id: 0 };  
  
    $scope.remove = function (capacidades) {  
        Capacidades.remove(capacidades);  
    }  
  
})
```

```
.factory('Capacidades', function () {  
    var capacidades = [  
        {  
            id: 0,  
            name: 'Vertical Cylinder1',  
            lastText: 'Vertical cylinder capacity used by a fluid',  
            image: 'images/tanque_vert.png',  
            variables: {  
                d:0,  
                l:0,  
                h:0,  
                volumen:0,  
                capacidad:0,  
                b:0,  
                id:0,  
                idCasing:0,  
                odTubing:0,  
                volumenAnular:0  
            }  
        }  
    ]  
})
```

Al momento de seleccionar una de las opciones para realizar los cálculos, la vista utiliza el controlador “CalculoCapacidadesCtrl”, este controlador se encarga de realizar la

lógica del cálculo y transformación de unidades, para la transformación de unidades se colocaron “watchers” sobre los datos de ingreso para que realicen el cálculo inmediato y transformen los datos a su unidad base, todos los resultados retornan en valores equivalentes a las unidades seleccionadas.

```

$scope.$watch('medidas[1].seleccionado', function(newValues, oldValues,$scope) {
  $scope.unidadVolumenNew=newValues;
  $scope.unidadVolumenOld=oldValues;

  for(var i=0; i<$scope.capacidades.length;i++)
  {
    $scope.capacidades[i].variables.volumen=Medidas.transformarIda($scope.capacidades[i].variables.volumen, $scope.unidadVolumenNew,$scope.unidadVolumenOld,'volumen');

    $scope.capacidades[i].variables.volumenAnular=Medidas.transformarIda($scope.capacidades[i].variables.volumenAnular, $scope.unidadVolumenNew,$scope.unidadVolumenOld,'volumen');

    $scope.capacidades[i].variables.volumen=$filter('setDecimal')($scope.capacidades[i].variables.volumen,3);

    $scope.capacidades[i].variables.volumenAnular=$filter('setDecimal')($scope.capacidades[i].variables.volumenAnular,3);
  }
});

```

Los cálculos se realizan en tiempo real, mientras se ingresan los datos la aplicación detecta que cuenta con los necesarios y realiza el cálculo automáticamente, se implementaron los algoritmos desarrollados por el área matemática de acuerdo a las instrucciones de la misma área, se agregaron funciones para transformación de unidades y formato de decimales como en el siguiente cálculo.

```

switch (capacidadID){

  case 0:{

    $scope.capacidad.variables.volumen=Medidas.transformarBase(((Math.PI * Math.pow(d,2)*h)/(4*(5.61458284))),idUnidadVolumen,'volumen');

    $scope.capacidad.variables.volumen=$filter('setDecimal')(Medidas.transformarIda($scope.capacidad.variables.volumen,

```

```

$scope.unidadVolumenNew,$scope.unidadVolumenOld,'volumen'),3);

    $scope.capacidad.variables.capacidad=$filter('setDecimal')((h*100)/(l),3);
    break;
}

```

Al final del desarrollo se entrega el siguiente prototipo funcional, listo para pruebas.

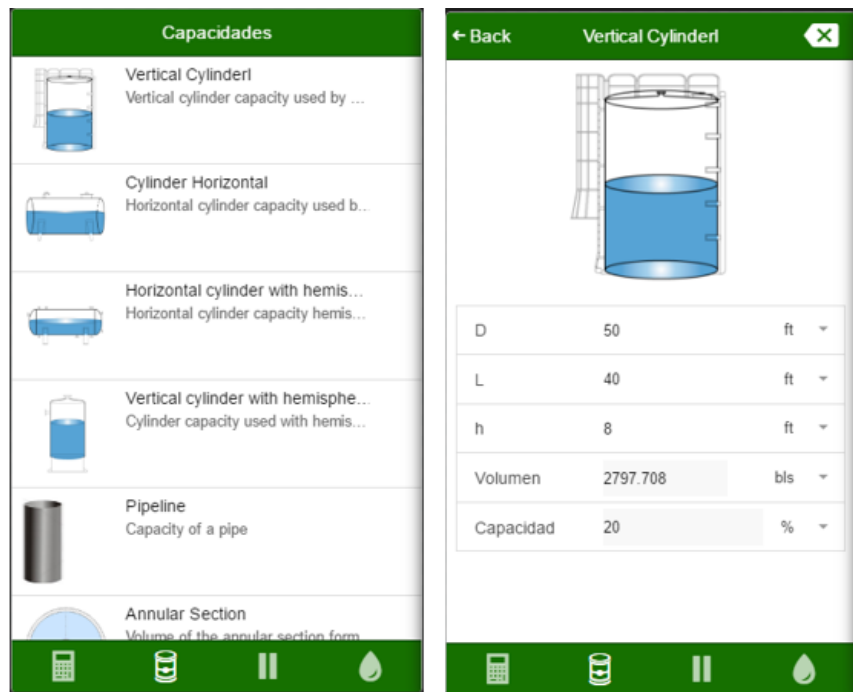


Ilustración 36 Prototipo Calculo volumen de capacidades captura 1

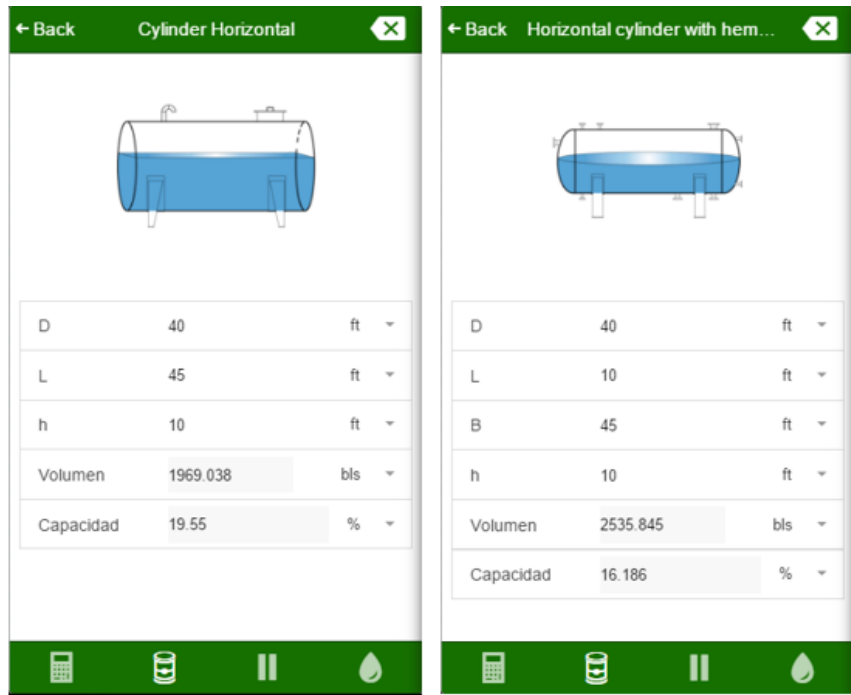


Ilustración 37 Prototipo Calculo volumen de capacidades captura 2

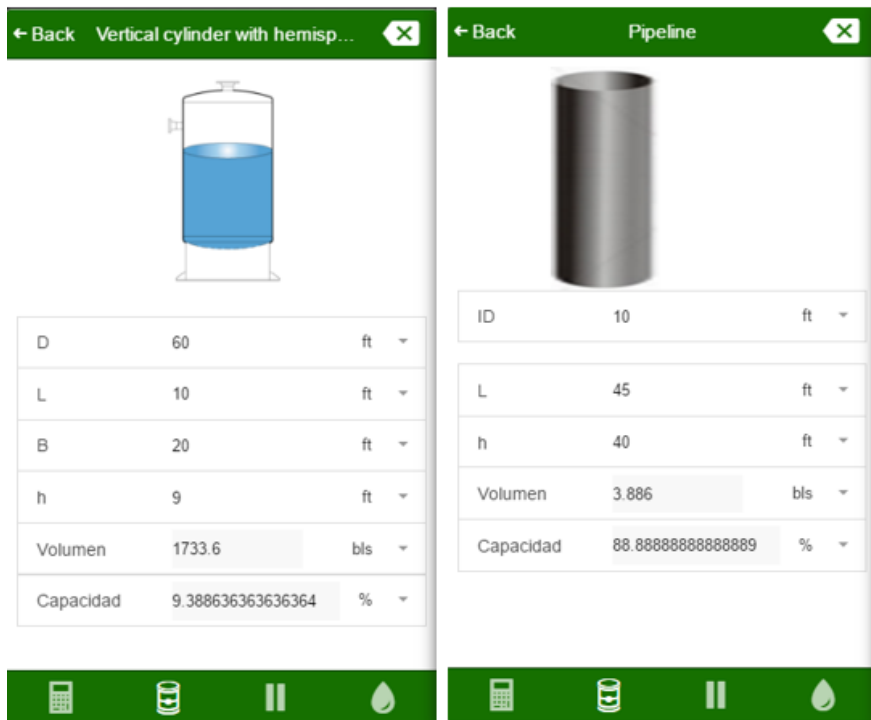


Ilustración 38 Prototipo Calculo volumen de capacidades captura 3

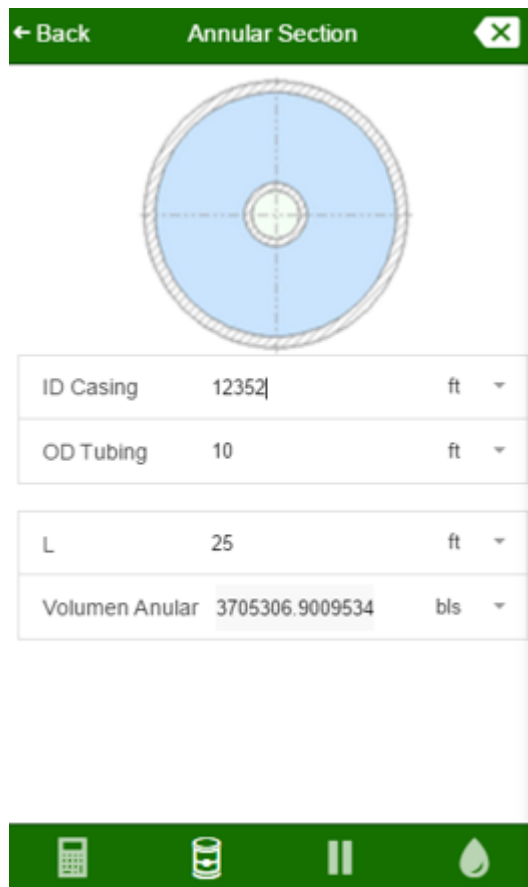


Ilustración 39 Prototipo Calculo volumen de capacidades captura 4

- *Pruebas*

Test	Acción	Resultado Esperado	Resultado Actual	Pasa la prueba
1	Calculo cilindro Vertical	D:50(ft) L: 40(ft) H: 8(ft) Vol: 2797.708 Capacidad: 20%	D:50(ft) L: 40(ft) H: 8(ft) Vol: 2797.708 Capacidad: 20%	Si
2	Calculo cilindro Horizontal	D:40(ft) L: 45(ft) H: 10(ft) Vol: 1969.038 Capacidad: 19.55%	D:40(ft) L: 45(ft) H: 10(ft) Vol: 1969.038 Capacidad: 19.55%	Si
3	Calculo	D:40(ft)	D:40(ft)	Si

	cilindro horizontal con hemisferios	L: 10(ft) B: 45 (ft) H: 10(ft) Vol: 2535.845 Capacidad: 16.186%	L: 10(ft) B: 45 (ft) H: 10(ft) Vol: 2535.845 Capacidad: 16.186%	
4	Calculo cilindro vertical con hemisferios	D:60(ft) L: 10(ft) B: 20 (ft) H: 9(ft) Vol: 1733.6 Capacidad: 9.38%	D:60(ft) L: 10(ft) B: 20 (ft) H: 9(ft) Vol: 1733.6 Capacidad: 9.38863636363664%	No (el resultado es correcto, pero se muestran más de 3 decimales)
5	Pipeline	ID:10(ft) L: 45ft) H: 40(ft) Vol: 3.886 Capacidad: 88.888888%	Ft:8 Cm:243.84 M: 243.84 Km: 0.002	No (el resultado es correcto, pero se muestran más de 3 decimales)
6	Sección Anular	ID:120(ft) OD: 10(ft) L: 25(ft) Vol: 347.28	ID:120(ft) OD: 10(ft) L: 25(ft) Vol: 347.28	No (el resultado es correcto, pero se muestran más de 3 decimales)

Las pruebas no fueron aprobadas ya que los valores de los resultados no estaban homologados, esto se debía a que no se aplicó el filtro al resultado de las capacidades, posteriormente se aplicaron las correcciones.

- *Reunion final de iteración I*

Durante la reunión se actualizaron los status de las tareas, adicionalmente el área de pruebas recomendó revisar las transformaciones de unidades ya que al momento de transformar y regresar al valor anterior este no era el mismo sino un valor aproximado en decimales, el área matemática explico que esto se debía a la exactitud de las fórmulas de transformación de unidades.

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F006	Filtro que redondea un valor en 3 decimales	Este filtro tiene la finalidad de homologar el número de decimales para presentar todos los resultados.	Alta	2	100%	
d	Transformación unidades para variables de ingreso	Esta funcionalidad permite realizar cálculos en diferentes unidades	Muy Alta	6	100%	
F008	Creación de vistas de capacidades	Creación de la vista de menú de capacidades y vista de una capacidad específica.	Alta	8	100%	
F0010	Cálculos de capacidades	Utilizar las formulas del área matemática para realizar los cálculos y posteriormente presentar en pantalla.	Alta	14	100%	

3.10.3. Iteración N°3

- *Reunión inicial de iteración*

En la reunión de inicio de iteración el equipo realiza la revisión de los requerimientos del módulo de diccionario de tuberías, este módulo no emplea ningún tipo de cálculo, únicamente realiza búsquedas de acuerdo a una serie de selecciones de datos

sucesivas, esta información se la tiene dentro de un archivo de Excel.

ESPECIFICACIONES API DEL TUBING										
DIÁMETRO NOMINAL	DIÁMETRO OD	GRADO	PESO			DIÁMETRO ID	DIÁMETRO DRIFT	COLLAPSE RESIST	PIPE BODY INTERNAL YIELD	VOLUMEN
			Non-upset	Upset	Int. It.					
in	in		lbs/ft			in	in	psi	psi	bl/100 ft
3/4	1.050	H-40	1.140	1.200		0.824	0.730	7680.000	7530.000	0.066
	1.050	J-55	1.140	1.200		0.824	0.730	10560.000	10360.000	0.066
	1.050	L-80	1.140	1.200		0.824	0.730	15370.000	15070.000	0.066
	1.050	N-80	1.140	1.200		0.824	0.730	15370.000	15070.000	0.066
	1.050	C-90	1.140	1.200		0.824	0.730	17290.000	16950.000	0.066
	1.050	T-95	1.140	1.200		0.824	0.730	18250.000	17890.000	0.066
	1.050	H-40	1.540			0.742	0.648	10010.000	10270.000	0.053
	1.050	J-55	1.540			0.742	0.648	13770.000	14120.000	0.053
	1.050	L-80	1.540			0.742	0.648	20020.000	20530.000	0.053
	1.050	N-80	1.540			0.742	0.648	20020.000	20530.000	0.053
	1.050	C-90	1.540			0.742	0.648	22530.000	23100.000	0.053
	1.050	T-95	1.540			0.742	0.648	23780.000	24380.000	0.053
	1.050	P-110	1.540			0.742	0.648	27530.000	28230.000	0.053

Los encargados del desarrollo deben realizar una estructura de datos adecuada para la consulta de esta información. Para esta iteración se definieron las siguientes tareas:

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F0010	Trasladar información de tuberías de Excel a una estructura de datos (JSON)	Esta tarea consiste en trasladar la información que fue receptada en un archivo de Excel hacia una estructura de datos de tipo JSON, esto con el fin de que la aplicación pueda realizar las consultas.	Muy Alta	8	0%	
F0011	Creación de vista para la consulta de tuberías	Consiste en la creación de la vista que tendrá el módulo de consulta de capacidades.	Alta	3	0%	
F0012	Funcionalidad de la vista de consulta de tuberías.	Consiste en realizar la funcionalidad de los controles de la vista, con el fin de que el usuario pueda realizar una búsqueda	Alta	4	0%	

- *Diseño de interfaz*

Para el diseño de la interfaz el departamento definió que la búsqueda este orientada en una serie de selecciones, para los que se planteó el siguiente esquema:

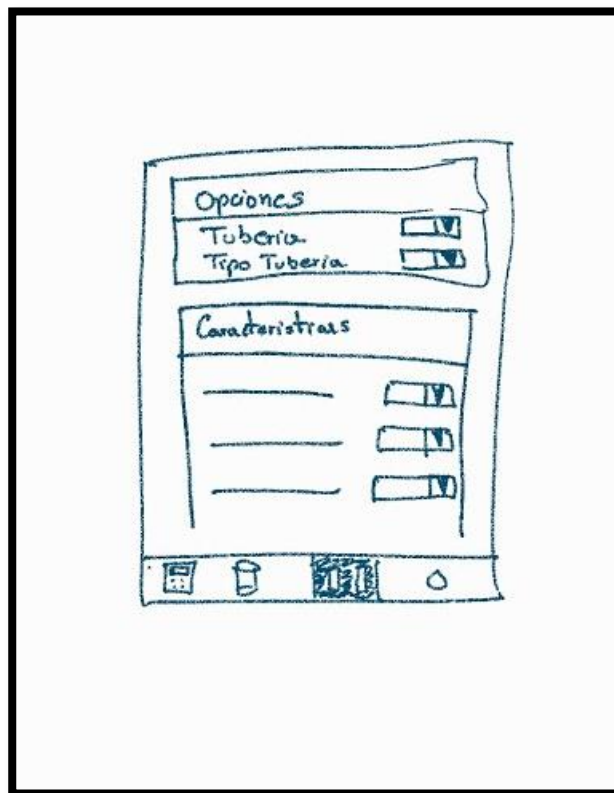


Ilustración 40 Diseño modulo consulta de tuberías

- *Desarrollo modulo*

El punto más importante en la etapa de Desarrollo para este módulo, consistía en estructurar la data de las tuberías, el esquema utilizado fue el siguiente:

```
var opcionTuberia=[
  {
    id:0,
    name:"Casing",
    tipoTuberia:[
      {
        id:0,
        name:"TAMSA",
        características:{
          dexterno:null,
          peso: null,
          grado:null
        }
      },{
        id:1,
        name:"API",
        características:{
          dexterno:null,
          peso: null,
          grado:null
        }
      }
    ]
  },{
    id:1,
    name:"Tubing",
    tipoTuberia:[
      {
        id:0,
        name:"TAMSA",
        características:{
          dexterno:null,
          peso: null,
          grado:null
        }
      },{
        id:1,
        name:"API",
        características:{
          dexterno:null,
          peso: null,
          grado:null
        }
      }
    ]
  }
]
```

Se asignó una estructura principal para los dos distintos tipos de tuberías los cuales posteriormente permitirán identificar sus pertenecientes dentro del catálogo de Api y Tamsa

```
var casingAPI = [  
  {  
    "nod":4.5,  
    "nid":4.09,  
    "peso":9.5,  
    "grado":"H-40",  
    "drift":3.965,  
    "resist":2760,  
    "yield":3190,  
    "volumen":1.625  
  },  
  {  
    "nod":4.5,  
    "nid":4.09,  
    "peso":9.5,  
    "grado":"J-55",  
    "drift":3.965,  
    "resist":3310,  
    "yield":4380,  
    "volumen":1.625  
  },  
]
```

```
var tubingTAMSA = [  
  {  
    "nod":1.05,  
    "nid":0.824,  
    "peso":1.14,  
    "grado":"H-40",  
    "drift":0.73,  
    "resist":7680,  
    "yiel":7530,  
    "volumen":0.066  
  },  
  {  
    "nod":1.05,  
    "nid":0.824,  
    "peso":1.14,  
    "grado":"J-55",  
    "drift":0.73,  
    "resist":10560,  
    "yiel":10360,  
    "volumen":0.066  
  },  
]
```

cada arreglo de cada tipo contiene el catálogo completo de las tuberías, este catálogo se encuentra dentro de un Factory de

Angular Js, el Factory esta denominado “CasingTubing” y se encuentra dentro del archivo “www/js/services.js”.

La vista de este modulo es muy sencilla, únicamente carga controles de tipo select los cuales de acuerdo a la selección iran actualizando las demás selecciones hasta que se muestren el resultado de la tubería buscada.

```
<ion-view view-title="Casing-Tubing">
  <ion-content class="padding">
    <div class="list card">
      <div class="item item-divider-platform item-positive">
        Opciones
      </div>
      <div class="item item-input item-select">
        <span class="input-label">Tuberia</span>

        <select ng-model="tuberiaSeleccionada" ng-options="tuberia.id as
tuberia.name for tuberia in opcionTuberia " ng-
change="setCombos(tuberiaSeleccionada,tipotuberia,seldexterno,selpeso,selgrado)"
></select>
      </div>
      <div class="item item-input item-select">
        <span class="input-label">Tipo de Tuberia</span>

        <select ng-model="tipotuberia" ng-options="tipoTuberia.id as
tipoTuberia.name for tipoTuberia in
opcionTuberia[tuberiaSeleccionada].tipoTuberia " ng-
change="setCombos(tuberiaSeleccionada,tipotuberia,seldexterno,selpeso,selgrado)"
></select>
      </div>
    </div>
    <div class="list card">
      <div class="item item-divider-platform item-positive">
        Caracteristicas
      </div>
      <div class="item item-input item-select">
        <span class="input-label">Nominal OD (in)</span>

        <select ng-model="seldexterno" ng-options="tuberia for tuberia in
dexterno " ng-
change="setCombos(tuberiaSeleccionada,tipotuberia,seldexterno,selpeso,selgrado)"
></select>

      </div>
      <div class="item item-input item-select">
        <span class="input-label">Weight (lbs/ft)</span>
        <select ng-model="selpeso" ng-options="peso for peso in peso " ng-
change="setCombos(tuberiaSeleccionada,tipotuberia,seldexterno,selpeso,selgrado)"
></select>
      </div>
    </div>
  </div>
</ion-content>
</ion-view>
```

```

<div class="item item-input item-select">
  <span class="input-label">Grade</span>
  <select ng-model="selgrado" ng-options="grado for grado in grado " ng-
change="setCombos(tuberiaSeleccionada,tipotuberia,seldexterno,selpeso,selgrado)"
></select>

</div>

<div class="item ">
  <span class="input-label" style="position: absolute">Nominal ID
(in)</span>
  <input type="tel" ng-model="nid" disabled="true" style="text-align: right;
float: right ">

</div>
<div class="item " >
  <span class="input-label" style="position: absolute">Diameter drift</span>
  <input type="tel" ng-model="drift" disabled="true" style="text-align:
right; float: right ">

</div>
<div class="item ">
  <span class="input-label" style="position: absolute">Collapse resist
(psi)</span>
  <input type="tel" ng-model="resist" disabled="true" style="text-align:
right; float: right ">
</div>
<div class="item ">
  <span class="input-label" style="position: absolute">Internal yield pressure
(psi)</span>
  <input type="tel" ng-model="yield" disabled="true" style="text-align:
right; float: right ">

</div>
<div class="item ">
  <span class="input-label" style="position: absolute">Volume (bl/100
ft)</span>
  <input type="tel" disabled="true" ng-model="volumen" style="text-align:
right; float: right ">
</div>
</div>
</ion-content>
</ion-view>

```

Toda la lógica de esta vista la maneja el controlador denominado “CasingTubingCtrl”, este controlador entre sus principales funciones tiene la capacidad de ir filtrando los resultados de acuerdo a las selecciones realizadas de datos anteriores, esta función se encarga de ir cambiando las opciones de los combos

```

$scope.setCombos= function(tuberiaSeleccionada,tipotuberia,dexterno,peso,grado)
{
    if (tuberiaSeleccionada == 0 && tipotuberia == 0)
        var base = $scope.dbcasingTAMSA;
    if (tuberiaSeleccionada == 0 && tipotuberia == 1)
        var base = $scope.dbcasingAPI;

    if (tuberiaSeleccionada == 1 && tipotuberia == 0)
        var base = $scope.dbtubingTAMSA;
    if (tuberiaSeleccionada == 1 && tipotuberia == 1)
        var base = $scope.dbtubingAPI;

    $scope.dexterno=base().distinct("nod")
    $scope.peso= base({ nod:dexterno }).distinct("peso");
    $scope.grado=base({ nod:dexterno},{ peso:peso }).distinct("grado");

    var item=base({ nod:dexterno},{ peso:peso},{ grado:grado}).get()[0];

    if(item==undefined)
    {
        $scope.nid=null;
        $scope.drift=null;
        $scope.resist=null;
        $scope.yield=null;
        $scope.volumen=null;
    }
    $scope.nid=item.nid;
    $scope.drift=item.drift;
    $scope.resist=item.resist;
    $scope.yield=item.yield;
    $scope.volumen=item.volumen;
}

```

Como resultado final, el prototipo para este módulo quedo de la siguiente manera:

Casing-Tubing	
Opciones	
Tuberia	Casing ▾
Tipo de Tuberia	API ▾
Características	
Nominal OD (in)	4.5 ▾
Weight (lbs/ft)	9.5 ▾
Grade	H-40 ▾
Nominal ID (in)	4.09
Diameter drift	3.965

Ilustración 41 Prototipo Consulta de características de tuberías Casing y Tubing

- *Pruebas*

Las pruebas de este módulo se las realizaron inmediatamente con las personas de pruebas, al tratarse únicamente de cargar información de acuerdo a selecciones, este módulo había sido implementado ya antes en otro proyecto del departamento por lo que el resultado estaba garantizado.

- *Reunión final de iteración*

Durante esta reunión el equipo realice un análisis del estatus del Proyecto, en el cual se concluyó que la mayoría de lo planificado había

sido cumplida, salvo que, durante el periodo de desarrollo, actividades de otros proyectos retrasaron la entrega de este módulo.

El equipo tenía una preocupación en cuanto a mantener actualizada la base de datos de estas tuberías por lo que desarrollo explico que la arquitectura de la aplicación permitía en cualquier momento tomar información de un servicio web. Posteriormente se procedió a actualizar el cuadro de actividades y tareas.

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F0010	Trasladar información de tuberías de Excel a una estructura de datos (JSON)	Esta tarea consiste en trasladar la información que fue receptada en un archivo de Excel hacia una estructura de datos de tipo JSON, esto con el fin de que la aplicación pueda realizar las consultas.	Muy Alta	8	100%	Tubo un retraso de 48 horas por atender otros proyectos
F0011	Creación de vista para la consulta de tuberías	Consiste en la creación de la vista que tendrá el módulo de consulta de capacidades.	Alta	3	100%	Tubo un retraso de 48 horas por atender otros proyectos
F0012	Funcionalidad de la vista de consulta de tuberías.	Consiste en realizar la funcionalidad de los controles de la vista , con el fin de que el usuario pueda realizar una búsqueda rápida e intuitiva	Alta	4	100%	Tubo un retraso de 48 horas por atender otros proyectos

3.10.4. Iteración N°4

- *Reunión inicial de iteración*

Para el inicio de esta iteración el equipo realiza la revisión de los requerimientos a implementar, este módulo comprende el cálculo de IPRS, para estos cálculos la aplicación debe permitir

calcularlos utilizando tres modelos matemáticos distintos, estos cálculos tienen como finalidad generar curvas en un gráfico por lo que la aplicación debe permitir generar graficas a partir de los resultados.

Este módulo implementa cálculos matemáticos por lo que el área encargada entrega los algoritmos desarrollado en JavaScript, para este desarrollo se receptaron los siguientes cálculos:

```
function ip_flu_lin(cau_tst, pre_res, pre_ffl_tst){
    return cau_tst / (pre_res - pre_ffl_tst);
}

function ip_flu_vog(cau_tst, pre_res, pre_ffl_tst){
    return 1.8 * cau_tst / (pre_res * (1 -
0.2*(pre_ffl_tst/pre_res) - 0.8 *
Math.pow((pre_ffl_tst/pre_res),2)))
}

function cal_cau_lin(ip, pre_res, pwf){
    return ip * (pre_res - pwf)
}

function cal_cau_vog(pre_res, pwf, qmax_flu){
    return qmax_flu * (1 - 0.2 * (pwf / pre_res) - 0.8 *
Math.pow((pwf / pre_res),2));
}

function cau_max_vog(pre_res, pwf, caudal){
    return caudal / (1 - 0.2 * (pwf / pre_res) - 0.8 *
Math.pow((pwf / pre_res),2))
}
```

Estos algoritmos requieren las siguientes variables:

```
var pre_res (Presión del reservorio)
var Pb (Presion de burbuja)
var pre_ffl_tst (Presión de fondo fluente de prueba)
var cau_tst (caudal)
var bsw (Corte de agua)
var IP_flu (IP del fluido)
var Qmax_flu (Caudal máximo del fluido)
var IP_oil (IP del petróleo)
var Qmax_oil (Caudal máximo del petróleo)
```

A continuación, se detallan cada uno de los cálculos para los modelos solicitados. Cada uno de ellos implementan las funciones básicas mencionados anteriormente-

IPR Lineal

```
IP_flu = ip_flu_lin(cau_tst, pre_res, pre_ffl_tst);
Qmax_flu = cal_cau_lin(ipfluido, pre_res, 0);
IP_oil = ip_flu_lin(Qo, pre_res, pre_ffl_tst);
Qmax_oil = cal_cau_lin(ippetroleo, pre_res, 0)
```

IPR Vogel

```
IP_flu = ip_flu_vog(cau_tst, pre_res, pre_ffl_tst);
Qmax_flu = cau_max_vog(pre_res, pre_ffl_tst, cau_tst);
IP_oil = ip_flu_vog(Qo, pre_res, pre_ffl_tst);
Qmax_oil = cau_max_vog(pre_res, pre_ffl_tst, Qo);
```

IPR Vogel Compuesto

```
var Q_flu_pb = 0;
var Q_oil_pb = 0;

if (pre_res > Pb) {
  if (pre_ffl_tst > Pb) {
    IP_flu = ip_flu_lin(cau_tst, pre_res, pre_ffl_tst)
    IP_oil = ip_flu_lin(Qo, pre_res, pre_ffl_tst)
  }
  else
  {
    IP_flu = cau_tst / (pre_res - Pb + Pb / 1.8 * (1 - 0.2 * (pre_ffl_tst / Pb) -
0.8 * Math.pow((pre_ffl_tst / Pb), 2)))
    IP_oil = Qo / (pre_res - Pb + Pb / 1.8 * (1 - 0.2 * (pre_ffl_tst / Pb) - 0.8
* Math.pow((pre_ffl_tst / Pb), 2)));
  }
}
else {
  IP_flu = ip_flu_vog(cau_tst, pre_res, pre_ffl_tst)
  IP_oil = ip_flu_vog(Qo, pre_res, pre_ffl_tst)
}

if (pre_res > Pb) {
  Q_flu_pb = cal_cau_lin(IP_flu, pre_res, Pb)
  Q_oil_pb = cal_cau_lin(IP_oil, pre_res, Pb)
  Qmax_flu = Q_flu_pb + IP_flu * Pb / 1.8
  Qmax_oil = Q_oil_pb + IP_oil * Pb / 1.8
}
else {
  Qmax_flu = cau_max_vog(pre_res, pre_ffl_tst, cau_tst)
  Qmax_oil = cau_max_vog(pre_res, pre_ffl_tst, Qo)
}
```

Para este módulo se definieron las siguientes tareas

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F0013	Creación de las vistas para el módulo de cálculos de IPR	Esta tarea consiste en la creación de las vistas HTML para este modulo	Muy Alta	5	0%	
F0014	Funcionalidad de las vitas del módulo de cálculos de IPR	Consiste en la implementación de los cálculos IPR, la carga de la opción de cálculos y transformación de unidades	Alta	12	0%	
F0015	Creación de modal para presentación de gráficos IPR	El cálculo de IPR necesita expresar los resultados de modo grafico para lo cual se creara un modal que recepte la información y genere un grafico	Alta	8	0%	

- *Diseño de interfaz*

Para el diseño de este módulo se continúa usando el esquema de una lista para los distintos tipos de cálculos presentando una imagen representativa al cálculo, el título principal del cálculo y una descripción corta.

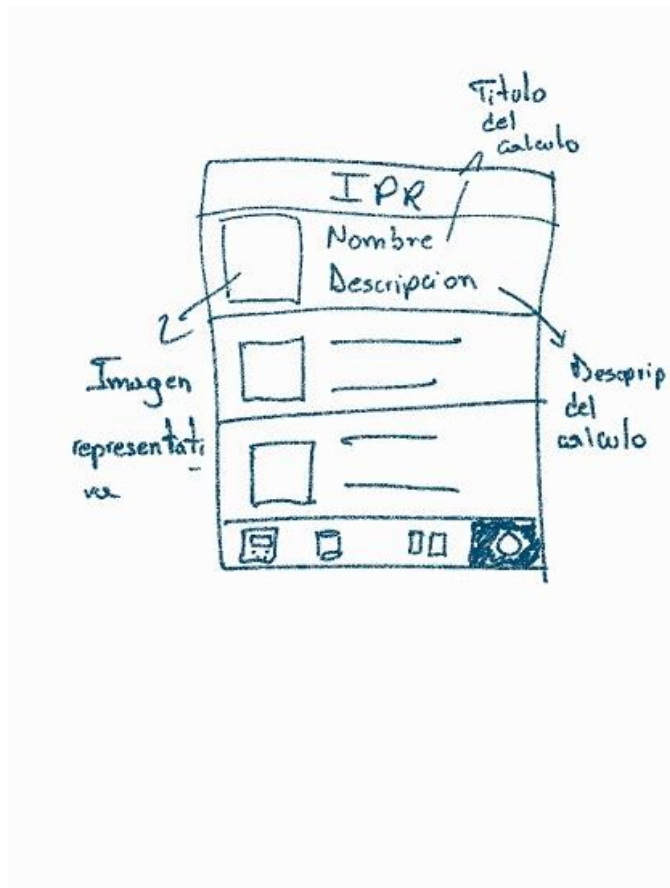


Ilustración 42 Diseño módulo cálculos de IPR

Al momento de seleccionar un tipo de cálculo, se presenta una vista en la cual tiene dos partes, la de datos de ingreso y la de resultados, una vez realizados los cálculos la aplicación debe tener la opción de graficar el resultado.

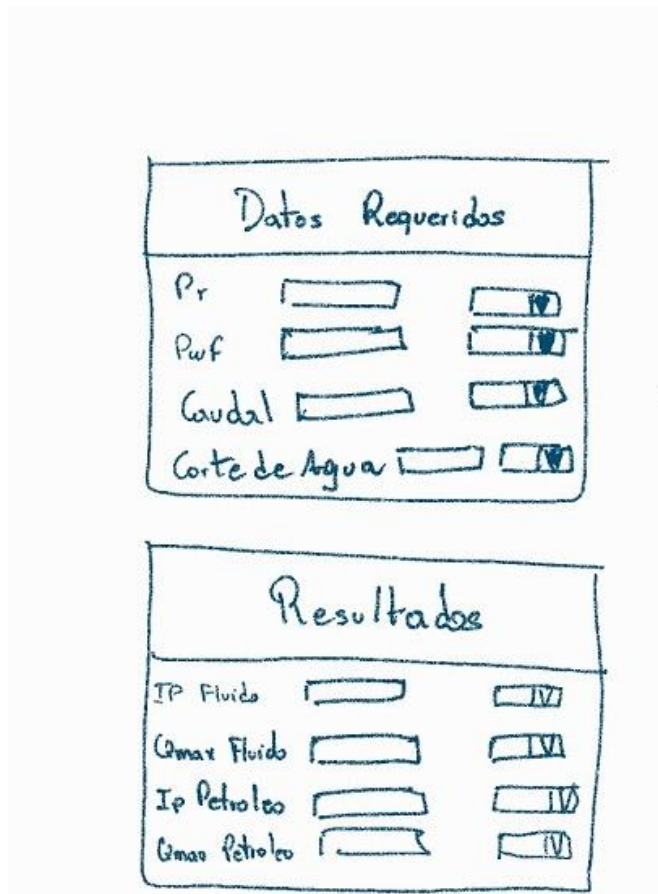


Ilustración 43 Diseño del módulo de cálculo IPR (Calculo específico)

- *Desarrollo modulo*

El Desarrollo de este módulo empezó por la creación de las vistas, la vista del menu de opciones está ubicada en el path “www/templates/tab-ipr.html”, esta vista carga el catálogo de cálculos IPR.

```

<ion-view view-title="IPR">
  <ion-content>
    <ion-list>
      <ion-item class="item-remove-animate item-icon-right item-thumbnail-left"
ng-repeat="ipr in iprs" type="item-text-wrap" href="#/tab/iprCalculos/{{ipr.id}}">
        <img src={{ipr.image}} style="background:#ffff; width:80px;
height:80px;">
        <h2>{{ipr.name}}</h2>
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>

```

```

        <p>{{ ipr.lastText }}</p>
        <i class="icon ion-chevron-right icon-accessory"></i>
    </ion-item>
</ion-list>
</ion-content>
</ion-view>

```

Una vez seleccionado un tipo de cálculo, se envía la opción seleccionada a la vista “www/templates/ipr-calculos.html”, esta vista carga los campos de ingreso de datos como resultados dependiendo de la opción seleccionada, esta vista funciona para cualquier tipo de cálculo seleccionada para el menu.

```

<ion-view view-title="{{ ipr.name }}">
    <ion-nav-buttons side="right"> <!-- right -->
        <button menu-toggle="left" class="button button-icon icon ion-backspace" ng-
click="borrar()"></button>
    </ion-nav-buttons>

    <ion-content class="padding">

        <div class="list card">

            <div class="item item-divider-platform item-positive">
                Datos requeridos
            </div>
            <div class="item item-input item-select">
                <span class="input-label">Pr</span>
                <input type="number" ng-model="ipr.variables.pr">
                <select ng-model="medidas[0].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidas[0].unidades " ></select>
            </div>

            <div class="item item-input item-select">
                <span class="input-label">Pwf</span>
                <input ng-model="ipr.variables.pwf" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
                <select ng-model="medidas[0].seleccionado" ng-options="unidad.id as

```

```

unidad.name for unidad in medidas[0].unidades " ></select>
</div>

<div ng-show="ipr.id==2" class="item item-input item-select">
  <span class="input-label">Pb</span>
  <input ng-model="ipr.variables.pb" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
    <select ng-model="medidas[0].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidas[0].unidades " ></select>
  </div>

<div class="item item-input item-select">
  <span class="input-label">Caudal</span>
  <input ng-model="ipr.variables.caudal" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
    <select ng-model="medidas[3].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidas[3].unidades " ></select>
  </div>

<div class="item item-input item-select">
  <span class="input-label">Corte de agua</span>
  <input ng-model="ipr.variables.corteagua" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
    <select>
      <option selected>%</option>

    </select>
  </div>
</div>
</div>

<div class="list card">
  <div class="item item-divider-platform item-positive">
    Resultados
  </div>
  <span class="input-label" >IP Fluido</span>

```

```

<div class="item item-input item-select">
  <input ng-model="ipr.variables.ipfluido" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
  <select ng-model="medidasExt[0].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidasExt[0].unidades " ></select>
</div>
<span class="input-label">Qmax Fluido</span>

<div class="item item-input item-select">
  <input ng-model="ipr.variables.qfluido" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
  <select ng-model="medidas[3].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidas[3].unidades " ></select>
</div>
<span class="input-label">IP petróleo</span>

<div class="item item-input item-select">
  <input ng-model="ipr.variables.ippetroleo" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
  <select ng-model="medidasExt[0].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidasExt[0].unidades " ></select>
</div>
<span class="input-label">Qmax petróleo</span>

<div class="item item-input item-select">
  <input ng-model="ipr.variables.qpetroleo" type="number" ng-
change="calcular(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt
[0].seleccionado)" >
  <select ng-model="medidas[3].seleccionado" ng-options="unidad.id as
unidad.name for unidad in medidas[3].unidades " ></select>
</div>
</div>

<button ng-model="habilitado" ng-disabled="habilitarGrafico()" class="button

```

```

button-full button-energized" ng-
click="graficar(ipr.id,medidas[0].seleccionado,medidas[3].seleccionado,medidasExt[0
].seleccionado);modal.show()">
    Graficar
  </button>
</ion-content>
</ion-view>

```

La funcionalidad de estas vistas la manejan dos controladores, "IprCtrl" es el controlador encargado de cargar las opciones de cálculos, "IprCalculosCtrl" implementa todos los cálculos Iprs además de la transformación de unidades para los campos de este módulo. Los dos controladores se encuentran dentro del archivo "www/js/controllers.js".

Las funciones otorgadas por el are matemática se encuentran dentro de este controlador. Estas funciones van a ser usadas en cada uno de los cálculos disponibles según corresponda.

```

function ip_flu_lin(cau_tst, pre_res, pre_ffl_tst){
  return cau_tst / (pre_res - pre_ffl_tst);
}
function ip_flu_vog(cau_tst, pre_res, pre_ffl_tst){
  return 1.8 * cau_tst / (pre_res * (1 - 0.2*(pre_ffl_tst/pre_res) - 0.8 *
Math.pow((pre_ffl_tst/pre_res),2)))
}

function cal_cau_lin(ip, pre_res, pwf){
  return ip * (pre_res - pwf)
}
function cal_cau_vog(pre_res, pwf, qmax_flu){
  return qmax_flu * (1 - 0.2 * (pwf / pre_res) - 0.8 * Math.pow((pwf / pre_res),2));
}

function cau_max_vog(pre_res, pwf, caudal){
  return caudal / (1 - 0.2 * (pwf / pre_res) - 0.8 * Math.pow((pwf / pre_res),2))
}

```

Para el cálculo de IPR Lineal se ejecuta el siguiente calculo:

```

case 0: {
  /*
  IPR LINEAL
  */
  IP_flu = ip_flu_lin(cau_tst, pre_res, pre_ffl_tst);
  Qmax_flu = cal_cau_lin($scope.ipr.variables.ipfluido, pre_res, 0);
  IP_oil = ip_flu_lin(Qo, pre_res, pre_ffl_tst);
  Qmax_oil = cal_cau_lin($scope.ipr.variables.ippetroleo, pre_res, 0)
  $scope.ipr.variables.ipfluido = IP_flu;
  $scope.ipr.variables.qfluido = Qmax_flu;
  $scope.ipr.variables.ippetroleo = IP_oil;
  $scope.ipr.variables.qpetroleo = Qmax_oil;

  $scope.ipr.variables.ipfluido=MedidasExt.transformarIda($scope.ipr.variables.ipfluido
, $scope.unidadIPNew,$scope.unidadIPOld,'ip');

  $scope.ipr.variables.ippetroleo=MedidasExt.transformarIda($scope.ipr.variables.ippet
roleo, $scope.unidadIPNew,$scope.unidadIPOld,'ip');
  $scope.ipr.variables.qfluido=Medidas.transformarIda($scope.ipr.variables.qfluido,
$scope.unidadCaudalNew,$scope.unidadCaudalOld,'caudal');

  $scope.ipr.variables.qpetroleo=Medidas.transformarIda($scope.ipr.variables.qpetrole
o, $scope.unidadCaudalNew,$scope.unidadCaudalOld,'caudal');
  $scope.ipr.variables.ipfluido=$filter('setDecimal')($scope.ipr.variables.ipfluido, 3);

  $scope.ipr.variables.ippetroleo=$filter('setDecimal')($scope.ipr.variables.ippetroleo,
3);
  $scope.ipr.variables.qfluido=$filter('setDecimal')($scope.ipr.variables.qfluido,3);

  $scope.ipr.variables.qpetroleo=$filter('setDecimal')($scope.ipr.variables.qpetroleo,3)
;
  break;
}

```

Para el cálculo de IPR de Vogel se ejecuta el siguiente cálculo.

```

case 1:{
  /*
  IPR VOGEL
  */
  IP_flu = ip_flu_vog(cau_tst, pre_res, pre_ffl_tst);
  Qmax_flu = cau_max_vog(pre_res, pre_ffl_tst, cau_tst);
  IP_oil = ip_flu_vog(Qo, pre_res, pre_ffl_tst);
  Qmax_oil = cau_max_vog(pre_res,pre_ffl_tst, Qo);

  $scope.ipr.variables.ipfluido = IP_flu;
  $scope.ipr.variables.qfluido= Qmax_flu;
  $scope.ipr.variables.ippetroleo = IP_oil;
  $scope.ipr.variables.qpetroleo = Qmax_oil;

  $scope.ipr.variables.ipfluido=MedidasExt.transformarIda($scope.ipr.variables.ipfluido
, $scope.unidadIPNew,$scope.unidadIPOld,'ip');

  $scope.ipr.variables.ippetroleo=MedidasExt.transformarIda($scope.ipr.variables.ippet
roleo, $scope.unidadIPNew,$scope.unidadIPOld,'ip');

  $scope.ipr.variables.qfluido=Medidas.transformarIda($scope.ipr.variables.qfluido,

```

```

$scope.unidadCaudalNew,$scope.unidadCaudalOld,'caudal');

$scope.ipr.variables.qpetroleo=Medidas.transformarIda($scope.ipr.variables.qpetroleo, $scope.unidadCaudalNew,$scope.unidadCaudalOld,'caudal');

    $scope.ipr.variables.ipfluido=$filter('setDecimal')($scope.ipr.variables.ipfluido, 3);

$scope.ipr.variables.ippetroleo=$filter('setDecimal')($scope.ipr.variables.ippetroleo, 3);
    $scope.ipr.variables.qfluido=$filter('setDecimal')($scope.ipr.variables.qfluido,3);

$scope.ipr.variables.qpetroleo=$filter('setDecimal')($scope.ipr.variables.qpetroleo,3)
;
    break;
}

```

Finalmente, para el cálculo de Vogel Compuesto se ejecuta el siguiente código.

```

case 2:
{
    /*
    IPR VOGEL COMPUESTO
    */
    var Q_flu_pb = 0;
    var Q_oil_pb = 0;

    if (pre_res > Pb) {
        if (pre_ffl_tst > Pb) {
            IP_flu = ip_flu_lin(cau_tst, pre_res, pre_ffl_tst)
            IP_oil = ip_flu_lin(Qo, pre_res, pre_ffl_tst)
        }
        else
        {
            IP_flu = cau_tst / (pre_res - Pb + Pb / 1.8 * (1 - 0.2 * (pre_ffl_tst / Pb) - 0.8 *
Math.pow((pre_ffl_tst / Pb), 2)))
            IP_oil = Qo / (pre_res - Pb + Pb / 1.8 * (1 - 0.2 * (pre_ffl_tst / Pb) - 0.8 *
Math.pow((pre_ffl_tst / Pb), 2)));
        }
    }
    else {
        IP_flu = ip_flu_vog(cau_tst, pre_res, pre_ffl_tst)
        IP_oil = ip_flu_vog(Qo, pre_res, pre_ffl_tst)
    }

    if (pre_res > Pb) {
        Q_flu_pb = cal_cau_lin(IP_flu, pre_res, Pb)
        Q_oil_pb = cal_cau_lin(IP_oil, pre_res, Pb)
        Qmax_flu = Q_flu_pb + IP_flu * Pb / 1.8
        Qmax_oil = Q_oil_pb + IP_oil * Pb / 1.8
    }
    else {

        Qmax_flu = cau_max_vog(pre_res, pre_ffl_tst, cau_tst)
        Qmax_oil = cau_max_vog(pre_res, pre_ffl_tst, Qo)
    }
}

```

```

$scope.ipr.variables.ipfluido = IP_flu;
$scope.ipr.variables.qfluido = Qmax_flu;
$scope.ipr.variables.ippetroleo = IP_oil;
$scope.ipr.variables.qpetroleo = Qmax_oil;

$scope.ipr.variables.ipfluido =
MedidasExt.transformarIda($scope.ipr.variables.ipfluido, $scope.unidadIPNew,
$scope.unidadIPOld, 'ip');
$scope.ipr.variables.ippetroleo =
MedidasExt.transformarIda($scope.ipr.variables.ippetroleo, $scope.unidadIPNew,
$scope.unidadIPOld, 'ip');
$scope.ipr.variables.qfluido =
Medidas.transformarIda($scope.ipr.variables.qfluido, $scope.unidadCaudalNew,
$scope.unidadCaudalOld, 'caudal');
$scope.ipr.variables.qpetroleo =
Medidas.transformarIda($scope.ipr.variables.qpetroleo, $scope.unidadCaudalNew,
$scope.unidadCaudalOld, 'caudal');

$scope.ipr.variables.ipfluido=$filter('setDecimal')($scope.ipr.variables.ipfluido, 3);
$scope.ipr.variables.ippetroleo=$filter('setDecimal')($scope.ipr.variables.ippetroleo,
3);
$scope.ipr.variables.qfluido=$filter('setDecimal')($scope.ipr.variables.qfluido,3);
$scope.ipr.variables.qpetroleo=$filter('setDecimal')($scope.ipr.variables.qpetroleo,3)
;
break;
}

```

Todos estos cálculos necesitan ser representados gráficamente para lo que se creó un modal Ionic, el cual se genera a partir de una vista html denominada “www/templates/modal.html”

```

<ion-modal-view ng-controller="ModalCtrl">w
  <header class="bar bar-header bar-positive">
    <h1 class="title">Gráfica IPR {{ipr.name}}</h1>
    <div class="button button-clear" ng-click="modal.hide()"><span class="icon
ion-close"></span></div>
  </header>
  <ion-content>
    <highchart id="chart1" config="chartConfig" class="span10"></highchart>
  </ion-content>
</ion-modal-view>

```

Esta vista ocupa una directiva de “highchart”, la cuales una librería JavaScript especializada en la generación de gráficos estadísticos.

El controlador asignado para la funcionalidad del modal es “ModalCtrl”, este modal cumple con la función de iniciar el objeto “Highchart” y posteriormente generar el grafico a partir de los resultados por el cálculo, este modal genera la graficas de todos los cálculos del módulo IPR

```
$scope.chartConfig = {  
  
  chart: {  
    renderTo: 'Chart',  
    type: 'spline',  
    zoomType: 'x', // puede ser xy  
    marginRight: 130  
  },  
  
  plotOptions: {  
    series: {  
      marker: {  
        enabled: true,  
        symbol: 'circle',  
        radius: 1.5  
      }  
    }  
  },  
  title: {  
    text: "Gráfica IPR"+" "+ $scope.ipr.name  
  },  
  xAxis: {  
    title: {  
      text: 'Caudal ('+$scope.medidas[3].unidades[idUnidadCaudal].name +)'  
    }  
  },  
  yAxis: {  
    min: 0,  
    title: {  
      text: 'Presión ('+$scope.medidas[0].unidades[idUnidadPresion].name +)'  
    }  
  },  
  credits: {  
    enabled: true, //Créditos del gráfico  
    style: {  
      color: '#359536',  
      fontSize: '10px'  
    },  
    text: "Syal",  
    href: 'http://www.sertecpet.com.ec'  
  },  
  
  series : [{  
    name : 'IPR Fluido',  
    data : arrFluido,  
    tooltip: {  
      valueDecimals: 2  
    }  
  },  
  {
```

```

name : 'IPR Petroleo',
data : arrOil,
tooltip: {
  valueDecimals: 2
}
}]
}

```

Como resultado del desarrollo se entrega como prototipo el siguiente modulo.

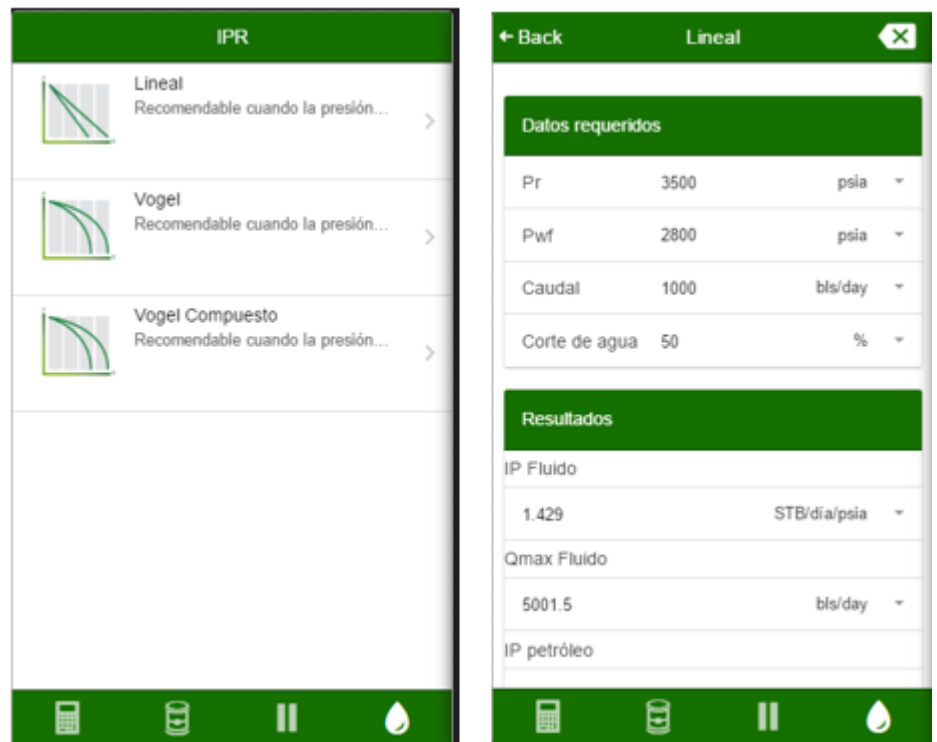


Ilustración 44 Prototipo módulo de cálculo IPR captural

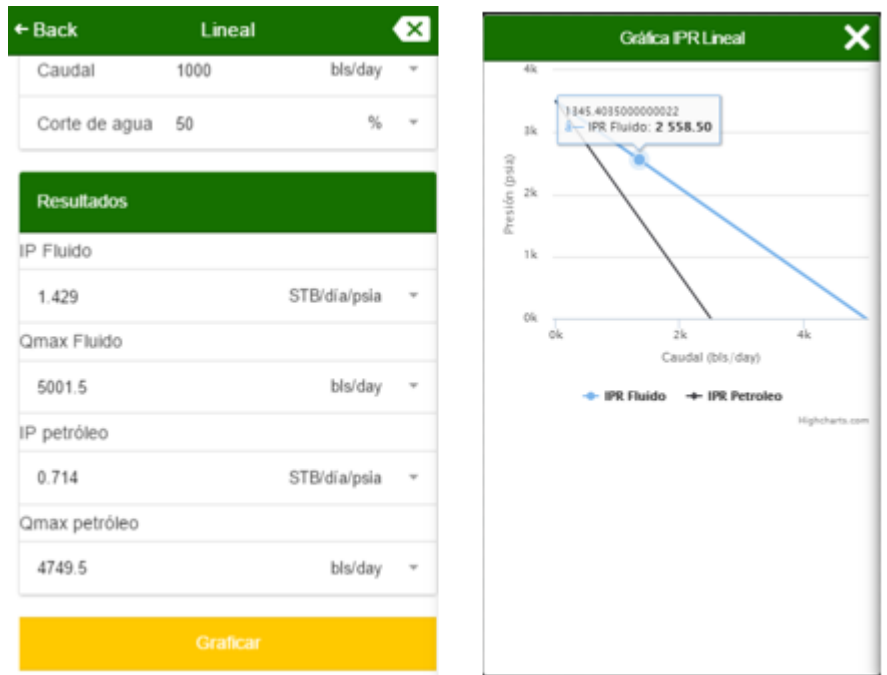


Ilustración 45 Ilustración 45 Prototipo módulo de cálculo IPR captura 2

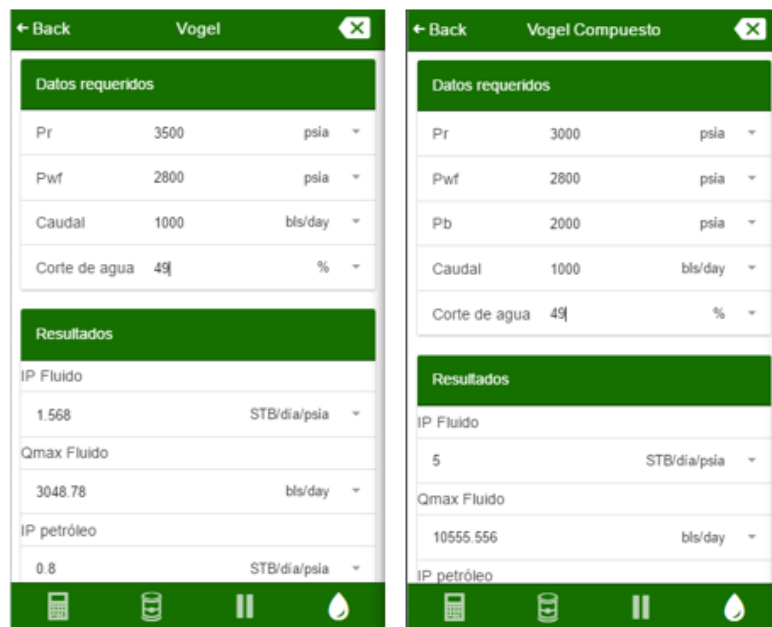


Ilustración 46 Ilustración 46 Ilustración 45 Prototipo módulo de cálculo IPR captura 3

- *Pruebas*

Test	Acción	Resultado Esperado	Resultado Actual	Pasa la prueba
1	Calculo IPR Lineal	IP Fluido 1.429(stb/dia/psia) Caudal maximo del fluido 5001.5 (bls/day) IP Petroleo 0.714 (stb/dia/psia) Caudal maximo del petroleo 4749.5 (bls/day)	IP Fluido 1.429(stb/dia/psia) Caudal maximo del fluido 5001.5 (bls/day) IP Petroleo 0.714 (stb/dia/psia) Caudal maximo del petroleo 4749.5 (bls/day)	Si
2	Calculo IPR Vogel	IP Fluido 1.568(stb/dia/psia) Caudal maximo del fluido 3048.78 (bls/day) IP Petroleo 0.8(stb/dia/psia) Caudal maximo del petroleo 1554.878 (bls/day)	IP Fluido 1.568 (stb/dia/psia) Caudal maximo del fluido 3048.78 (bls/day) IP Petroleo 0.8(stb/dia/psia) Caudal maximo del petroleo 1554.878 (bls/day)	Si
3	Calculo IPR Vogel Compuesto	IP Fluido 1.429(stb/dia/psia) Caudal maximo del fluido 5001.5 (bls/day) IP Petroleo 0.714 (stb/dia/psia) Caudal maximo del petroleo 4749.5 (bls/day)	IP Fluido 1.429(stb/dia/psia) Caudal maximo del fluido 5001.5 (bls/day) IP Petroleo 0.714 (stb/dia/psia) Caudal maximo del petroleo 4749.5 (bls/day)	Si

Todas las pruebas concernientes a los cálculos pasaron sin ningún problema, para la representación del grafico si surgieron problemas por lo que el área de Desarrollo se tomó cuatro horas más de lo presupuestada para el tiempo de pruebas, esto debido a que el grafico necesitaba una cantidad de decimales mayor a la que se enviaba para poder realizar el grafico.

O problema presentado durante las pruebas fue el ingreso de valores inadecuados, la aplicación caía en cálculos infinitos y se detenía su funcionamiento, para esto el área de pruebas sugirió la implementación de validaciones antes de generar un cálculo.

- *Reunión final de iteración*

En esta reunión se entregó el prototipo terminado, el área de Investigación y desarrollo realizo varias sugerencias respecto al diseño global de la interfaz, se determinó que el área de marketing realizara una evaluación acerca de este tema, se generó una acta de entrega del proyecto y se lo publico en Ionic View para que se puedan realizar presentaciones de la aplicación antes de publicarla.

Se actualizaron las tareas previstas para esta iteración.

ID	Característica	Descripción	Prioridad	Horas Estimadas	% Completado	Observaciones
F0013	Creación de las vistas para el módulo de cálculos de IPR	Esta tarea consiste en la creación de las vistas HTML para este modulo	Muy Alta	5	100%	
F0014	Funcionalidad de las vitas del módulo de cálculos de IPR	Consiste en la implementación de los cálculos IPR, la carga de la opción de cálculos y transformación de unidades	Alta	12	100%	Necesita desarrollar validaciones para evitar cálculos infinitos
F0015	Creación de modal para presentación de gráficos IPR	El cálculo de IPR necesita expresar los resultados de modo grafico para lo cual se creará un modal que recepte la información y genere un grafico	Alta	8	100%	Tomo más tiempo del esperado durante la etapa de pruebas

4. Conclusiones y recomendaciones

5. Bibliografía

TABLA DE CONTENIDOS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR	1
1. Introducción	2
1.1. Dedicatoria	3
1.2. Agradecimiento	4
1.3. Antecedentes	5
1.4. Justificación	6
1.5. Objetivos	7
1.5.1. Objetivo general	7
1.5.2. Objetivos específicos	7
2. Situación actual del desarrollo para móviles	8
2.1. Análisis de estado.....	8
2.2. Tipos de aplicaciones móviles.....	13
2.2.1. Aplicaciones nativas.....	14
2.2.2. Aplicaciones móviles web.....	16
2.2.3. Aplicaciones híbridas	17
2.2.4. Análisis de ventajas y desventajas	19
2.3. Aplicaciones empresariales	20
2.4. Metodología de diseño	21
2.4.1. Metodologías existentes	22
2.4.2. Agile modeling.....	23
2.5. Metodologías de desarrollo	26

2.5.1.	Metodologías existentes	27
2.5.2.	Análisis de ventajas y desventajas	29
2.5.3.	Prototyping	31
•	Análisis de ventajas y desventajas de prototyping	33
2.6.	HERRAMIENTAS DE DESARROLLO	34
2.6.1.	Herramientas disponibles	34
2.6.2.	Ionic app framework	38
2.6.3.	Cordova	51
3.	Diseño, desarrollo de la aplicación	53
3.1.	Solicitud y condiciones del proyecto	53
3.2.	Arquitectura propuesta para el desarrollo	53
3.3.	Designación de roles	53
3.4.	Requerimientos.....	54
3.5.	Descripción requerimientos:.....	54
3.6.	Tabla de requerimientos	55
3.7.	Planificación.....	56
3.8.	Diseño Base.....	58
3.8.1.	Pantalla Inicial (Splash)	58
3.8.2.	Esquema base de interacción	59
3.9.	Desarrollo Base	60
3.10.	Iteraciones.....	62
3.10.1.	Iteración N°1	62
3.10.2.	Iteración N°2	77
3.10.3.	Iteración N°3	92
3.10.4.	Iteración N°4	101
4.	Conclusiones y recomendaciones	119

5.	Bibliografía	119
6.	Anexos	121

6. Anexos

Ilustración 1	Evolución móviles.....	9
Ilustración 2	Evolución iphone	9
Ilustración 3	Canal online	11
Ilustración 4	App nativa	15
Ilustración 5	App web	17
Ilustración 6	App híbridas	19
Ilustración 7	Ciclo de vida	24
Ilustración 8	Transcurso	25
Ilustración 9	Modelo Prototipado	27
Ilustración 10	Modelo Espiral	28
Ilustración 11	Modelo Cascada	28
Ilustración 12	Modelo RAD.....	29
Ilustración 13	Etapas prototipado	33
Ilustración 14	PhoneGap	36
Ilustración 15	Xamarin.....	37
Ilustración 16	Archivos proyecto	41
Ilustración 17	Comandos en iOS y Android	42
Ilustración 18	HTML base ionic	43
Ilustración 19	Menú ionic	43
Ilustración 20	Codigos angularJs	44
Ilustración 21	Tareas asignadas	44
Ilustración 22	Venta modal	45
Ilustración 23	Codigos formularios.....	46
Ilustración 24	Codigos controlador	47

Ilustración 25	Contenedor central	48
Ilustración 26	Side menú.....	48
Ilustración 27	Arquitectura cordova.....	52
Ilustración 28	Pantall Inicial (Splash)	58
Ilustración 29	Esquema base interacción (Tabs9).....	59
Ilustración 30	Esquema base de interacción (Botones y funcionalidad).....	60
Ilustración 31	Estructura de archivos del proyecto	61
Ilustración 32	Diseño módulo conversión de unidades.....	65
Ilustración 33	Diseño módulo conversión de unidades (vista unitaria)	66
Ilustración 34	Diseño calculo volumen de capacidades (menú de cálculos) ...	82
Ilustración 35	Diseño calculo volumen de capacidades (calculo especifico) ...	83
Ilustración 36	Prototipo Calculo volumen de capacidades captura 1	88
Ilustración 37	Prototipo Calculo volumen de capacidades captura 2	89
Ilustración 38	Prototipo Calculo volumen de capacidades captura 3	89
Ilustración 39	Prototipo Calculo volumen de capacidades captura 4	90
Ilustración 40	Diseño modulo consulta de tuberias	94
Ilustración 41	Prototipo Consulta de características de tuberías Casing y Tubing.....	100
Ilustración 42	Diseño módulo cálculos de IPR	105
Ilustración 43	Diseño del módulo de cálculo IPR (Calculo específico)	106
Ilustración 44	Prototipo módulo de cálculo IPR captura1	115
Ilustración 45	Ilustración 45 Prototipo módulo de cálculo IPR captura 2	116
Ilustración 46	Ilustración 46 Ilustración 45 Prototipo módulo de cálculo IPR captura 3.....	116

