

**UNIVERSIDAD POLITÉCNICA DE PACHUCA**

**MEMORIA TÉCNICA DE ESTADÍA**

**DESARROLLO DE LA APLICACIÓN TUZO BUS APP**

PRESENTADO POR:

**DERAHIN ROJO FUNETES**

PARA OBTENER EL TÍTULO DE

**INGENIERO EN SOFTWARE**

ASESORADO POR:

**BARTOLOME FERNANDO MORENO VERA**

ASESOR ACADÉMICO

**ING. GELIER JONATHAN REYES VITE**

ASESOR TÉCNICO

# **AGRADECIMIENTOS**

*A mis padres*

# **RESUMEN**

Al día de hoy podemos encontrar en el mercado una gran cantidad de dispositivos móviles, los cuales son utilizados por personas de cualquier rango de edad, es una realidad que actualmente el *internet of things,* hace que cada dispositivo tenga una conexión a internet independiente, la cual no se limita por el tamaño del mismo, esto facilita y cubre las necesidades de los usuario para consultar información en línea.

En los siguientes párrafos de esta memoria de estadía se presenta el diseño y desarrollo de la aplicación “**TUZO BUS APP**”, en la cual se explica al usuario la manera en la que se desempeña dicho servicio de transporte, al ser un transporte público de nueva introducción al municipio, se decide el desarrollo de esta herramienta para facilitar el aprendizaje de dicho servicio en donde se puede consultar las diferentes rutas que existen dependiendo de la necesidad del usuario.

Se decide el desarrollo visual user experience donde se utiliza la normativa de diseño enfocado a la visualización del sistema operativo Android la cual es metodología visual desarrollada por google, donde se encuentra el lenguaje de diseño en el que predominan animaciones transiciones efectos de profundidad así como iluminación y sombras, El entorno de Material Design es un elemento muy importante que juega dentro de la aplicación ya que sobre esta herramienta se pretende captar una mayor atención en el entorno visual de la aplicación.

El desarrollo del proyecto se administra mediante una herramienta móvil llamada Trello, la cual ayudo a mantener el equipo comunicado y un orden en los proceso que se estaba desarrollando para tener una misma sintaxis permitiendo que el proceso fuera monitoreado por cualquier integrante del equipo y dividiendo los módulos en un panel de información los cuales eran, Planeación, Diseño, Desarrollo, Revisión y Finalizados

Es utilizada la metodología scrum la cual fue ágil y flexible para gestionar el desarrollo ya que fue aumentando el número de módulos durante el desarrollo los cuales eran inestables o cambiantes y esta metodología se adapta a los proyectos con requisitos inestables y para los que se requieren rapidez y flexibilidad la cual fue perfecta para el desarrollo de la aplicación móvil.

Falta hablar de Android studio

**RESUMEN**

RESUMEN 2

# **CAPÍTULO 1**

En esta memoria de estadía se presenta el proyecto “Tuzo bus App” , el cual es diseñado y desarrollado en oficinas de Creative Labs S. de R.L. de C. V, una empresa mexicana ubicada en la ciudad de Pachuca, Hidalgo, con dirección en calle Lázaro Cárdenas No. 100 Col. Adolfo López Mateos, C.P. 42094 la cual es innovadora en campos de realidad virtual, realidad aumentada así como ofreciendo servicios web, aplicaciones móviles realizando investigaciones de tecnologías de la información así como visión computacional ofreciendo su sello en cada uno de sus productos los cual son responsabilidad con el cliente así como la creatividad única en cada uno de sus proyectos.

En el año 2012 Xólotl creative labs decide crecer y posicionarse como una de las principales proveedoras de tecnología en el estado, comenzando con el desarrollo de videojuegos educativos adoptando nuevas tecnologías y diseños los cuales lograron captar la atención de clientes potenciales que ayudarían al crecimiento de la misma desafiándose cada vez más en proyectos los cuales conllevaban mas nivel de dificultad en el desarrollo.

En este mismo año la empresa desarrollo diversas campañas de Realidad aumentada mediante cámaras web así como Kinect para el seguimiento de objetos, con estas nuevas tecnologías logran desarrollos educativos mas interactivos, demostrando su trabajo y creatividad en ferias de tecnología como Expo Innova Universitaria, gracias a esto se logra la venta a uno de sus principales clientes Kronos innovación.

A principios de 2013 Xólotl creative labs decide aumentas sus capacidades de desarrollo invirtiendo tiempo en proyectos en el Game Engone de unity 3D permitiendo exportar sus proyectos a nivel multiplataforma de las más utilizadas tales como: Android, Ios. Los cuales son los principales sistemas operativos para dispositivos móviles, así como sistemas operativos de escritorio como Linux, Windows, Mac.

La empresa vio la oportunidad de presentarles a más personas su producto el cual constaba de una aplicación que ayudara al usuario a comprender como funcionaba el nuevo sistema de transporte público del municipio, y al mismo tiempo ayudar a creces las ventas de otros servicio de la empresa, realizando estudio de cuantas personas descargaban la aplicación así como el rango de edad de las usuarios para segmentar mercado en otros productos.

# **CAPÍTULO 2**

# Marco contextual

Actualmente se encuentran aplicaciones con poca tecnología y falta de experiencia de usuario sobre herramientas que ayuden al entendimiento de cómo funcionan los servicios de transporte público, siendo Pachuca una ciudad de crecimiento y de gran turismo donde hay que promover el servicio de transporte publico y educar a los nuevos usuarios sobre el uso correcto del mismo, aunque para personas que han utilizado servicios parecidos no sea necesario el conocimiento de las rutas, se incluirán publicaciones de información en este medio sobre el transporte público.

El nuevo servicio de transporte que ingresa a la ciudad de Pachuca, es innovador y organizado, y se encuentra la oportunidad del desarrollo de una aplicación móvil para el uso y conocimiento del mismo, ya que se tiene conocimiento de que el 85% de la población tiene un dispositivo móvil donde puede consultar la información que le sea de interés, de tal manera se plantea utilizar ese porcentaje de usuarios para abarcar a más clientes sobre el desarrollo de aplicaciones móviles de nueva generación, con más aspecto visual y experiencia de usuario más atractiva.

A base de esta modernización al sistema de transporte público, se pretende el desarrollo de una aplicación accesible a todos los usuarios, sin necesidad de una conexión a internet en todo momento, contando con modo offline, en la que el usuario pueda consultas las diferentes rutas y paradas que tiene el servicio de transporte Tuzo Bus con tanto con una interfaz amigable y con temas de accesibilidad para los usuarios mayores que tengan problemas de vista.

La aplicación Tuzo Bus App, es una aplicación para dispositivos móviles con sistema operativo Android, enfocada al sistema de transporte público de la ciudad de Pachuca.

## **Descripción de problema**

El nuevo sistema de transporte público de Pachuca, llega con un desconocimiento sobre su servicio, y retrasando a los usuarios a su destino por una desinformación del servicio así como de las rutas y las paradas que realiza en cada estación.

## **Objetivos**

El objetivo del proyecto es educar a la población en la forma de utilizar el servicio de transporte público para mejorar el tiempo en traslado, y su información acerca del sistema así como de conocer los puntos de ascenso y descenso de usuarios de igual manera se dará un aproximado en los tiempos de transporte de una estación a otra.

### **Objetivo General**

Desarrollo de una aplicación móvil utilizado en lenguaje de programación Java y Android Studio como entorno de desarrollo de la aplicación, y las herramientas de desarrollo de vida de software, para una correcta implementación en la evolución del proyecto, el cual facilitara a los usuarios en el conocimiento sobre cómo y dónde abordad las unidades de transporte público para llegar a su destino así como las correctas rutas del sistema de transporte metropolitano como sus horarios, conocer las rutas troncales y rutas alimentadoras que ofrece el servicio.

### **Objetivo específicos**

1. Identificar las limitaciones que tiene el usuario con el sistema de transporte público, y las áreas de oportunidad para el desarrollo.
2. Diseñar diagramas de casos de uso y de actividades dándole solución al problema.
3. Diseño y desarrollo de los módulos previamente evaluados.
4. Desarrollo y trazado de mapas con librería de google.
5. Diseño y desarrollo de la base de datos que se encargara de alimentar la información de la aplicación.

## **2.3** **Justificación**

El desarrollo de este proyecto surge de la innovación que pretende el área metropolitana ante un nuevo sistema de trasporte público el cual llega con desinformación por parte de los usuarios del mismo, de tal manera surge un área de oportunidad para el desarrollo de una aplicación móvil beneficiando así a los usuario para que puedan consultar las diferentes rutas que ofrece el sistema así como los horario y precios , contando con accesibilidad para abarcar a más usuarios adaptando el tamaño de letra y colores para un entorno visual más apropiado y cómodo dependiendo el tipo de usuario, así como con una sección de noticias donde se mantengan informados sobre los cambios repentinos o interrupciones del servicio.

La accesibilidad a las nuevas tecnologías, permite la correcta implementación de este proyecto ya que ahora se cuenta con una mejor conexión inalámbrica a las redes así como con dispositivos inteligentes capaces de soportar aplicaciones robustas haciendo una disponibilidad de las misma en cualquier lugar y a cualquier hora sin tener que estas físicamente frente a una computadora o una conexión obligada a datos móviles.

## **2.4. Alcances y Limitaciones**

La aplicación “Tuzo Bus” está enfocada a informar al usuario sobre el funcionamiento del servicio de trasporte público *Tuzobus*, encontrándose con un diseño atractivo y adaptable a la vista así como una fluidez en la experiencia de usuario en la cual podrán consultarse las rutas así como los horarios. Los alcances del proyecto son:

1. Los usuarios que utilicen la aplicación podrán consultar las rutas que deberán abordar para disminuir sus tiempos de traslado hacia su destino.
2. Los usuarios obtendrán la información necesaria acerca de horarios y los lugares donde hacen parada cada unidad del Tuzobus.
3. La aplicación contara con un módulo de noticias para que se mantengan informados sobre los cambios repentinos o interrupciones del servicio.
4. La aplicación está diseñada con diferentes temas para adaptar la experiencia visual en usuarios que no distingan a primera vista colores claros.
5. La aplicación esta optimizada para trabajar mejor con dispositivos Android 4.0 o superiores

Las limitaciones del proyecto son:

1. La aplicación solo es desarrollada para el sistema de transporte Tuzobus que se encuentra en la zona metropolitana de Hidalgo.
2. En su primer inicio se requiere una conexión a internet para la descarga de mapas necesarios para su correcto funcionamiento.
3. La aplicación solo se desarrolla bajo el sistema operativo Android

# **CAPÍTULO 3**

# Marco teórico

Dentro del proceso y desarrollo de la aplicación “Tuzo Bus” es necesario entender algunos términos así como las herramientas utilizadas que fueron parte fundamental para el correcto desarrollo de dicha aplicación tales como: Material Design, Experiencia de usuario, Servicios Web basados en REST y JSON, Fragments, Actividades, Versión de Android, Librerías de google, Uml, por mencionar los más importantes.

## 3.1 Sistema operativo para dispositivos móviles

Los sistema operativos de la actualidad se diseñan visualmente más atractivos y simplificados para el usuario aunque cuantas con más opciones y configuraciones que los SO anteriores los grandes sistemas se han enfocado en hacer su interfaz más agradable para el usuario, para otorgar así un funcionamiento más fluido y darle al usuario lo que él requiera con menos procedimientos o instrucciones, de igual manera estos SO se enfocan en las conexiones inalámbricas como las redes 4G LTE así como redes WiFI, la existencia de estos servicios hace más completo un dispositivo móvil al grado de que un usuario normal puede realizar actividades que comúnmente realizaría en una computadora personal eliminando con cada actualización limitaciones que pudieran limitar actividades.

Los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas.

Estas capas son:

Kernel: Es el que proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece distintos servicios a las superiores como son los controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.

Middleware: Es el conjunto de modulos que hacen posible la propia existencia de aplicaciones para moviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajeria y comunicaciones, codecs multimedia, interpretes de paginas web, gestion de sispositivos y seguridad.

Entorno de ejecucion de aplicaciones: consiste en un gestor de aplicaciones y un conjunto de interfaces programables abiertas para facilitar la creacion de software.

Interfaz de usuario: Facilita la interaccion con el usuario y el diseño de la presentacion visual de la aplicacion. Los servicios que incluye son el de componentes graficos (botones, pantallas, listas,etc) y el del marco de interaccion.

Los sistemas operativos moviles mas usados son:

### Android

El sistema operativo el cual es el líder del mercado con una actualización constante debido a estas actualizaciones es el sistema operativo mas fragmentado hasta el momento teniendo 15 versiones publicadas desde el 2008 hasta la fecha siendo la última versión *Oreo* con Android 8.0 manejando un nivel de API 26, Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes y televisores. Gracias a esto y a su agradable interfaz visual cuenta con una cuota de mercado cercana al 85% el sistema operativo de Google se caracteriza por ser abierto y disponible para cualquier fabricante interesando en utilizarlo para sus dispositivos móviles.

### iOS

Es un sistema operativo más exclusivo y con más restricciones, ya que no permite la modificación de sus equipos lo cual asegura la integridad y la capacidad de proteger los datos del usuario ante algún robo de información haciendo sus quipos muy seguros, ofreciendo siempre una experiencia más estable y segura, Por otro lado este sistema está limitado solo a sus unidades y no se puede instalar a empresas de terceros

### Windows Phone

Microsoft que está realizando un gran esfuerzo financiero para posicionar Windows Phone como una tercera opción interesante para los consumidores después de que llegara tarde a la fiesta de los smartphones. Su alianza con Nokia y su posterior compra le ha ayudado a darse a conocer mejor e ir arañando cuota de mercado a los dos líderes. Los últimos datos hablan de un 2,5% a nivel mundial.

Windows Phone destaca por su pantalla de inicio personalizable que ofrece las notificaciones de las apps de una manera sencilla y limpia. Además ofrece una experiencia de usuario muy buena independientemente del tipo y gama de terminal en que se esté usando.

### Firefox OS

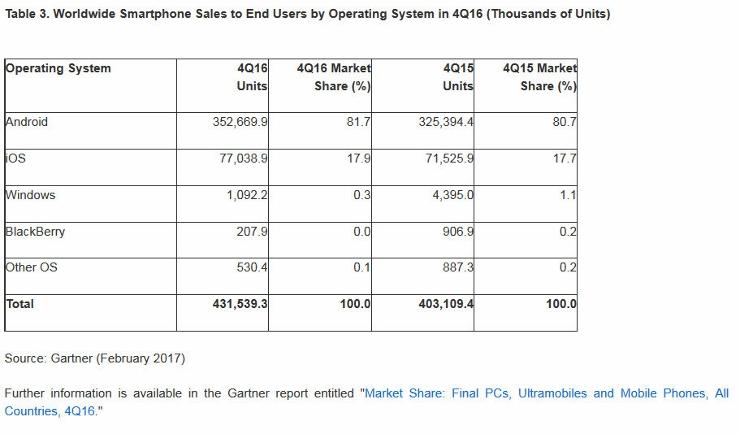
Un sistema operativo basado en HTML5 con núcleo Linux, de código abierto. Desarrollado por Mozilla Corporatión con apoyo de empresas como Telefónica. El sistema operativo está basado en Linux y usa la tecnología de Mozilla, Gecko. Se basa en estándares abiertos como por ejemplo HML5, CSS3 y JavaScript.

Conclusiones

Como conclusión puedo decir que todos los sistemas operativos móviles tienen sus ventajas y desventajas, actualmente Android ocupo gran parte del mercado debido a que sus aplicaciones tienen un entorno amigable y a su vez muchos usuarios pueden desarrollar sus propias aplicaciones, iOS por otro lado es más estable y presenta menos vulnerabilidades, Windows phone así mismo tiene una interfaz amigable, los demás sistemas operativos quizá menos utilizados pero dependiendo de la necesidad del usuario brindan sus respectivas ventajas.

## 3.2. Mercado de los sistemas operativos móviles

El mercado de sistemas operativos (OS) móviles ha sido una carrera de sólo dos caballos desde hace tiempo, con la plataforma de Google, Android, en la delantera, seguido de Apple iOS. Eso es más cierto que nunca ahora que [Gartner ha publicado data](http://www.gartner.com/newsroom/id/3609817" \t "_blank) que apunta a que estas dos plataformas combinadas representan el 99.6% de la participación del mercado total.



## Sistema Operativo Android

En julio de 2005, Google adquirió Android Inc., una pequeña compañía de Palo Alto, California fundada en 2003. En aquel entonces, poco se sabía de las funciones de Android Inc. fuera de que desarrollaban software para teléfonos móviles. Esto dio pie a rumores de que Google estaba planeando entrar en el mercado de los teléfonos móviles.

En Google, el equipo liderado por Rubin desarrolló una plataforma para dispositivos móviles basada en el [núcleo Linux](https://es.wikipedia.org/wiki/N%C3%BAcleo_Linux) que fue promocionado a fabricantes de dispositivos y operadores con la promesa de proveer un sistema flexible y actualizable. Se informó que Google había alineado ya una serie de fabricantes de hardware y software y señaló a los operadores que estaba abierto a diversos grados de cooperación por su parte.

### 3.3.1 Arquitectura de Android

Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma. En el siguiente diagrama se muestran los componentes principales de la plataforma Android.

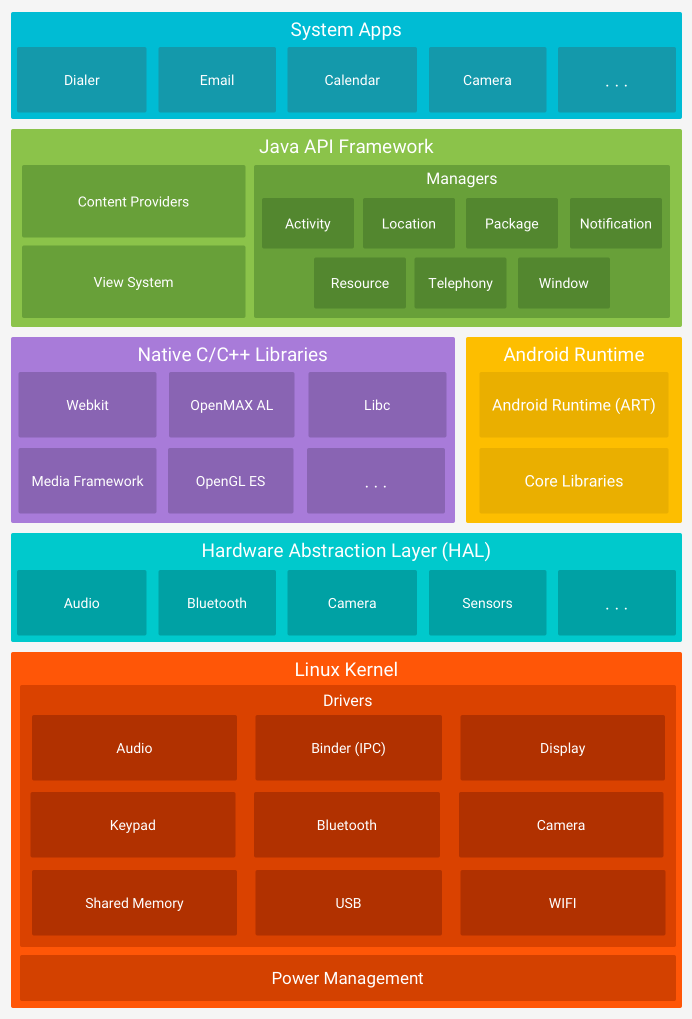


Figura 3.2: Arquitectura Android (Fuente https://developer.android.com/)

Apps del sistema: en Android se incluye un conjunto de apps centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elije instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Settings del sistema).

Framework de la Java API: todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares

Bibliotecas C/C++ nativas: muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la API del framework de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps.

Tiempo de ejecución de Android: para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del [tiempo de ejecución de Android (ART)](http://source.android.com/devices/tech/dalvik/index.html?hl=es-419). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como [Jack](https://source.android.com/source/jack.html?hl=es-419), y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Capa de abstracción de hardware (HAL): la [capa de abstracción de hardware (HAL)](https://source.android.com/devices/index.html?hl=es-419#Hardware%20Abstraction%20Layer) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al [framework de la Java API](https://developer.android.com/guide/platform/index.html?hl=es-419#api-framework) de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de [bluetooth](https://source.android.com/devices/bluetooth.html?hl=es-419).

Kernel de Linux: la base de la plataforma Android es el kernel de Linux. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocesos y la administración de memoria de bajo nivel.

### 3.3.2. Versiones

El historial de versiones del sistema operativo Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial (de prueba), Android 1.0, fue lanzada en septiembre de 2008 y la última hasta el momento en agosto del 2017 han pasado 10 años ya y hasta el momento son 10 versiones de Android cada una con el nombre de un postre en inglés y alfabéticamente. En la tabla 3.1 Versiones Android, encontraras un listado de las versiones publicadas con sus detalles

Tabla 3.1: Versión Android

|  |  |  |  |
| --- | --- | --- | --- |
| NOMBRE DE VERSION | VERSION ANDROID | FECHA DE LANZAMIENTO | API |
| Android 1.01​ | 1.0 | 23 de septiembre 2008 | 1 |
| Android 1.11​ | 1.1 | 9 de febrero 2009 | 2 |
| Cupcake | 1.5 | 27 de abril de 2009 | 3 |
| Donut | 1.6 | 15 de septiembre de 2009 | 4 |
| Eclair | 2.0–2.1 | 26 de octubre de 2009 | 5-7 |
| Froyo | 2.2–2.2.3 | 20 de mayo 2010 | 8 |
| Gingerbread | 2.3–2.3.7 | 6 de diciembre 2010 | 9–10 |
| Honeycomb2​ | 3.0–3.2.6 | 22 de febrero de 2011 | 11–13 |
| Ice Cream Sandwich | 4.0–4.0.5 | 18 de octubre 2011 | 14–15 |
| Jelly Bean | 4.1–4.3.1 | 9 de julio de 2012 | 16–18 |
| KitKat | 4.4–4.4.4, 4.4W–4.4W.2 | 31 de octubre de 2013 | 19–20 |
| Lollipop | 5.0–5.1.1 | 12 de noviembre de 2014 | 21–22 |
| Marshmallow | 6.0–6.0.1 | 5 de octubre de 2015 | 23 |
| Nougat | 7.0 - 7.1.2 | 15 de junio de 2016 | 24-25 |
| Oreo | 8.0 | 21 de agosto de 2017 | 26 |

## Arquitectura cliente-servidor

Este modelo Cliente/Servidor empezó a ser aceptado a finales de los 80’s. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado.

Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end [15].

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

• Administrar la interfaz de usuario.

• Interactuar con el usuario.

• Procesar la lógica de la aplicación y hacer validaciones locales.

• Generar requerimientos de bases de datos.

• Recibir resultados del servidor.

• Formatear resultados.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end [15].

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

• Aceptar los requerimientos de bases de datos que hacen los clientes.

• Procesar requerimientos de bases de datos.

• Formatear datos para trasmitirlos a los clientes.

• Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

## Servicios web basados librería OKHttp REST

HTTP es la herramienta moderna para las aplicaciones que necesitan conexión. Una forma para intercambiar datos y medios. HTTP te ayuda a cargar datos de una forma eficiente con mayor velocidad y ahorrando datos de trasferencia.

OkHttp es un cliente HTTP que ofrece:

* La compatibilidad con HTTP/2 permite que todas las solicitudes en el mismo host compartan un socket.
* La agrupación de conexiones reduce la latencia de las solicitudes (si HTTP/2 no está disponible).
* GZIP reduce el tamaño de las descargas.
* Con el cache se omiten consultas repetidas.

OkHttp persiste cuando la red es problemática: se recuperará silenciosamente de problemas de conexión comunes. Si su servicio tiene múltiples direcciones IP, OkHttp intentará direcciones alternativas si falla la primera conexión. Esto es necesario para IPv4 + IPv6 y para servicios alojados en centros de datos redundantes. OkHttp inicia nuevas conexiones con las características modernas de TLS (SNI, ALPN), y vuelve a TLS 1.0 si falla el protocolo de enlace.

## JSON

**JSON** (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del [Lenguaje de Programación JavaScript](http://javascript.crockford.com/), [Standard ECMA-262 3rd Edition - Diciembre 1999](http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf). JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituído por dos estructuras:

* Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un *objeto*, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
* Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o sequencias.

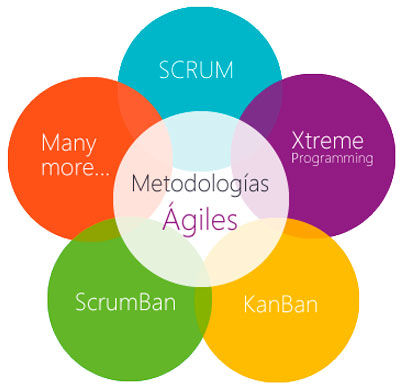
Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

## Desarrollo agil

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Teniendo gran importancia el concepto de "Finalizado" (Done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.



Doce principios de Agilidad

Principio 1: Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continuada de software con valor

El objetivo es lograr un cliente satisfecho, lo que contribuirá a que tengamos más clientes en el futuro. ¿Pero cómo? Proporcionando a los clientes la solución que realmente quieren, aunque ya sabemos que esto no es posible sin ser adaptativos, y sin entregas tempranas y continuas de software funcionando. Esta flexibilidad necesaria, aunque es posible en los ciclos de vida predictivos, resulta demasiado cara.

Principio 2: Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Empleando un ciclo de vida adaptable estamos abiertos al cambio ya que no existe ningún diseño inicial al que debamos ceñirnos cada vez que queramos realizar un cambio. Además, cualquier petición de cambio nos hará felices, pues será un paso más que nos permitirá acercarnos a lo que el cliente realmente desea.

Principio 3: Entregamos software funcional frecuentemente, entre dos semanas y un mes, con preferencia por periodos de tiempo lo más corto posibles.

El cliente tendrá una mejor comprensión de lo que quiere cuando vea el software en funcionamiento. Nosotros, recibiremos información (feedback) que podremos utilizar para adaptarlo.

Hay distintos marcos Agiles que emplean diferentes iteraciones. Por ejemplo, en Scrum no están permitidas las iteraciones de más de un mes, mientras que otros marcos Ágiles aceptan iteraciones más largas. Siempre y cuando sean suficientes para crear un incremento significativo (software en funcionamiento) preferiremos las iteraciones más cortas.

Principio 4: Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto

En un entorno predictivo la participación de la empresa/cliente se limita generalmente a especificar los requisitos al inicio, y de nuevo al final a aprobar la solución final. Sin embargo, en un entorno adaptable necesitamos que la empresa/cliente trabaje a diario con los desarrolladores, ya que sus inputs son la fuente de la adaptabilidad.

Principio 5: Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

Un entorno ágil se basa en un equipo multifuncional y auto-organizado que se auto-gestiona y encuentra su camino en lugar de recibir órdenes. Esta es una gran responsabilidad para los desarrolladores, y no todos son capaces de trabajar de esta manera.

Cuando tenemos los miembros de equipo adecuados, debemos confiar en ellos, motivarles y darles la posibilidad de permitir la agilidad

Principio 6: El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara

En un entorno tradicional los miembros del equipo se centran en sus actividades de especialista, incluso podrían estar ubicados en lugares diferentes, por lo general en sus respectivos departamentos dentro de la organización.

A veces ni siquiera podemos llamarlos “equipo”; no son más que una serie de personas que trabajan en el mismo proyecto. Por el contrario en un entorno Ágil necesitamos un verdadero equipo, en el que los miembros deben estar co-localizados para poder comunicarse continuamente. Nada puede sustituir una conversación cara a cara.

Aunque ciertamente es una gran ventaja contar con equipos co-localizados, esto no significa que no podamos tener un proyecto Ágil con un equipo “distribuido”. En estos casos sin embargo, necesitaremos aprovechar al máximo la tecnología para reducir al mínimo la falta comunicación cara a cara, y asumir un nivel de productividad inferior al final del día.

Principio 7: El software funcionando es la principal medida progreso

¡Un producto acabado al 99 % es un producto que esta 0 % “completo” o “hecho”!.

Pero entonces, ¿cómo conocer el progreso de nuestro trabajo sin entrar en detalles técnicos? Recuerda que estamos interesados en mantener al cliente involucrado en el proyecto, y para ello debemos tratar de evitar los detalles técnicos, y mantener un lenguaje sencillo, dado que en muchas ocasiones se tratará de un cliente “no técnico”.

La solución pasa por diferenciar los artículos del Backlog de Producto únicamente en dos categorías: “completo” y “no completo”. Esta simple distinción es suficiente ya que los elementos del Backlog son lo bastante pequeños para mostrar nuestro progreso simplemente diferenciando entre completo/no completo.

Principio 8: Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

Trabajar no es el objetivo; alcanzar el producto es el objetivo.

Podría parecernos que hacer horas extras puede acelerar las cosas, pero en realidad reduce los outputs disminuyendo la productividad y aumentando los defectos. Es preferible mantener un ritmo sostenido a lo largo del tiempo.

Principio 9: La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad

No tener un diseño inicial no significa que no tengamos que estar preocupado por el diseño.

Los proyectos ágiles tienen diseño, lo que ocurre es que este se realiza en cada iteración para cada elemento del Backlog de Producto.

Debemos prestar atención a la excelencia técnica y el buen diseño para evitar problemas; sin olvidar que el objetivo es encontrar una solución lo “suficientemente buena” más que la solución perfecta.

Principio 10: La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Un proyecto Ágil se gestiona y entrega de manera simple.

Por ejemplo, la gestión del alcance se realice simplemente detallando la información esencial en una tarjeta o nota adhesiva (ficha); no son necesarios instrumentos sofisticados para gestionar el producto. Además, hacerlo de manera sencilla favorece la colaboración del cliente.

Principio 11: Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

Por lo general la gente trabaja mejor cuando se siente respetada y están autorizados para decidir cómo funcionar.

Además, es mejor que todos los miembros del equipo sean responsables de todo el proyecto. Por ejemplo, sí los diseñadores no funcionan de manera aislada, entonces estarán constantemente en contacto con los programadores, y pueden utilizar la información que se genera para mejorar los diseños y hacerlos más prácticos.

Principio 12: A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

¡Creemos que siempre hay margen de mejora, sin importar lo bien que estemos haciendo las cosas!

Por ello necesitamos tiempo para investigar la iteración anterior y encontrar la manera de implementar mejoras, por muy pequeñas que sean. El objetivo es mejorar un poco cada iteración.

## Scrum

proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

## Diagramas del UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación

Diagramas del UML El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Diagramas del UML

Diagrama de Clases

Volver Los diagramas de clases describen la estructura estática de un sistema. Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. Un ejemplo puede ser la clase Aviones que tiene atributos como el modelo de avión, la cantidad de motores, la velocidad de crucero y la capacidad de carga útil. Entre las acciones de las cosas de esta clase se encuentran: acelerar, elevarse, girar, descender, desacelerar.

Diagrama de Objetos Volver Los Diagramas de Objetos están vinculados con los Diagramas de Clases. Un objeto es una instancia de una clase, por lo que un diagrama de objetos puede ser visto como una instancia de un diagrama de clases. Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases.

Diagrama de Casos de Uso Volver Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario. Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

Diagrama de Estados Volver En cualquier momento, un objeto se encuentra en un estado particular, la luz está encendida o apagada, el auto en movimiento o detenido, la persona leyendo o cantando, etc. . El diagrama de estados UML captura esa pequeña realidad.

Diagrama de Secuencias Volver Los diagramas de clases y los de objetos representan información estática. No obstante, en un sistema funcional, los objetos interactúan entre sí, y tales interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.

Diagrama de Actividades Volver Un diagrama de actividades ilustra la naturaleza dinámica de un sistema mediante el modelado del flujo ocurrente de actividad en actividad. Una actividad representa una operación en alguna clase del sistema y que resulta en un cambio en el estado del sistema. Típicamente, los diagramas de actividad son utilizados para modelar el flujo de trabajo interno de una operación.

Un **diagrama de paquetes** en el [Lenguaje Unificado de Modelado](https://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado) representa las dependencias entre los **paquetes** que componen un modelo. Es decir, muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones.

Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido.

Diagrama de actividades

Diagrama de casos de uso

Trello

De aquí para delante explicamos todos los servicios que se utilizaron así como herramientas

Minimo 15 paginas