

“Aplicación móvil de solicitud de cotizaciones”

Ricardo Antonio Márquez Enríquez ⁽¹⁾
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral(ESPOL)
Campus Gustavo Galindo, Km. 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
ricamarq@espol.edu.ec⁽¹⁾

Xavier Ochoa Chehab
Escuela Superior Politécnica del Litoral, Ph.D., xavier@cti.espol.edu.ec

Resumen

Las personas necesitan solicitar sus cotizaciones computacionales de forma rápida, barata y confiable. Se diseñó un Sistema de solicitud de cotizaciones para una empresa computacional, la cual corre en dispositivos móviles con sistema operativo Android. La aplicación ofrece la opción de poder ver que productos hay en stock y si los mismos cuentan con descuentos al comprar en mayor cantidad. Al ver el producto que necesita, el usuario podrá adherirlo al carrito de compras y después solicitarlo. Puede mostrar las cotizaciones pendientes y las que le han enviado, para que pueda ver en qué estado están y tener un respaldo al momento de comprar el producto para que el precio cotizado se cumpla. También le permite al usuario ver los servicios extras que ofrece la empresa, contactarse con la empresa y compartir la aplicación con cualquiera.

Palabras Claves: *Android, multimedia, computación.*

Abstract

People need their computational orders quickly, cheaply and reliably. We designed a request for price system for a computer company, which runs on mobile devices with Android operating. The application offers the option to see which products are in stock, and if they have discounts when buying in larger quantities. Seeing the product you need, you can attach it to shopping cart and then apply. You can see outstanding lists of prices and sent to you, so you can see in what state they are and have a backup when buying a product of the price list given. It also allows the user to see the extra services offered by the company, contact the company and share the application with everyone.

Key words: *Android, multimedia, computation.*

1. Análisis del problema

Aquí se ve cual es el problema, la justificación del porque se realiza este proyecto, las soluciones existentes hasta el momento y las herramientas que se usará para solucionar el problema previamente mostrado.

El problema principal al realizar una solicitud de cotización, es el de tener que llamar a la empresa, ir a la empresa o vía mail, sin saber si la empresa cuenta con tal producto. Al tener solo estas opciones, el cliente no cuenta con algo llamativo que le permita movilidad o que agilite el proceso.

Para resolver el problema antes mostrado, se decidió hacer una aplicación que ofrezca una opción diferente y de forma gratuita para clientes con dispositivos móviles, que le permita realizar una solicitud de cotización desde la comodidad del lugar donde se encuentre y permitiéndole además poder ver el stock de la empresa y si este cuenta con algún descuento, creando así un vínculo más cercano entre la empresa y el usuario. También se cuenta con una opción de mapas, para poder ubicar de mejor manera al usuario y poderlo dirigir hacia la empresa de ser necesario.

Recientemente han comenzado a aparecer aplicaciones móviles para las empresas computacionales de todo el mundo, sin embargo ninguna cumplía con todas las especificaciones que se encuentran en este proyecto. La mayoría son opciones para ver los productos que ofrecen las empresas de computación.

Existen varias plataformas móviles en el mercado, las cuales son Blackberry, Ios, Symbian, WIndows phone y Android, pero se decidió usar la plataforma Android, con lenguaje Java y base de datos MySQL, ya que la mayoría de celulares en el mundo tienen el sistema operativo Android embebido en ellos.

2. Análisis de la solución

Se realizó el levantamiento de requerimientos y se separó entre requerimientos funcionales y no funcionales. Se hizo los casos de uso para ver las interacciones posibles, se hizo la arquitectura de la solución y se llegó a definir hasta donde llegaría este proyecto.

El levantamiento de requerimientos en todo proyecto de software es importante, se debe hacer

reuniones constantes con el cliente para poder dejar según como el cliente quiera. Se debe enfatizarles que no todo se puede hacer y explicar cuáles son los límites como programador.

Los requerimientos funcionales se refieren a cómo debe comportarse la aplicación ante diferentes entradas y como debe funcionar en situaciones específicas y dar un servicio determinado. En ciertos casos estos requerimientos también son conocidos por mostrar que es lo que el programa no debería hacer.

Los requerimientos no funcionales no se refieren a las funciones en sí, se enfocan en la fiabilidad, velocidad de respuesta o almacenamiento del programa.

Los casos de uso nos sirven para especificar la comunicación y la interacción que hay en el sistema con los usuarios u otros sistemas, se basan en los requerimientos ya que muestran lo que fue descrito ahí.

Para el cliente se realizaron los siguientes casos de uso:

- Manda la solicitud de cotización seleccionando lo necesario.
- Busca los productos, su stock y descuentos existentes.
- Consulta últimas cotizaciones.
- Mira los servicios que ofrece la empresa.
- Comparte la aplicación con sus conocidos.
- Busca el local de la empresa en el mapa.
- Contacta a la empresa.

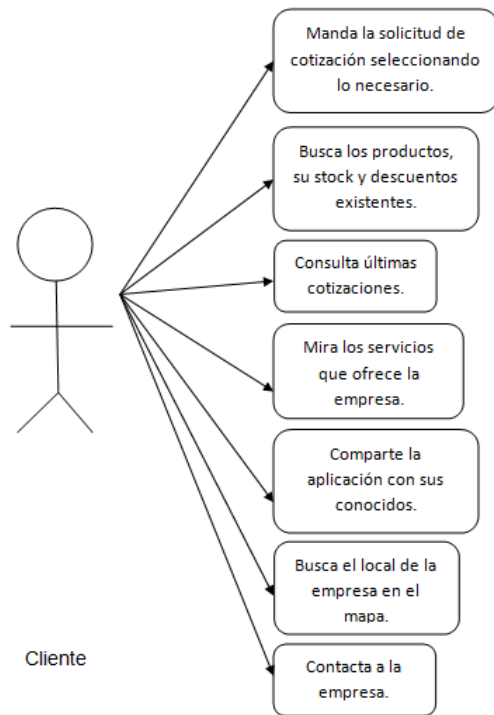


Figura 2.1 Caso de uso cliente

Para el usuario se realizaron los siguientes casos de uso:

- Recibe la solicitud de cotización.
- Envía y guarda la cotización.
- Recibe llamadas de clientes.
- Recibe correos de clientes.

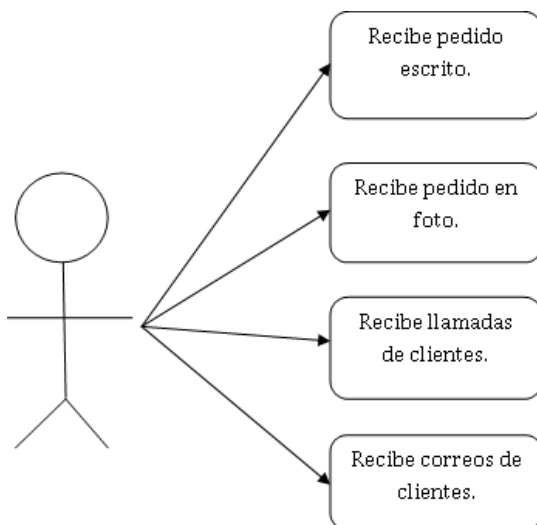


Figura 2.2 Caso de uso usuario

La arquitectura de la solución es un gráfico que representa la arquitectura base de la aplicación, para poder mostrar al cliente una versión gráfica de cómo quedará estructurada y facilitar su creación en la parte de diseño.

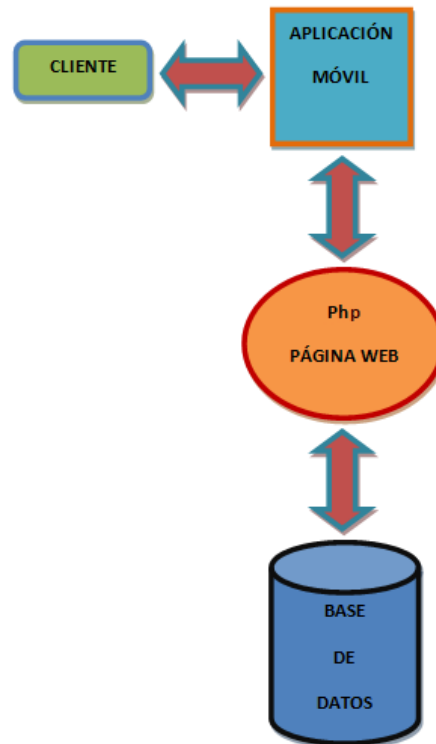


Figura 2.3 Arquitectura de la solución

El alcance es todo lo que se va a realizar, hasta donde va a llegar nuestra aplicación, que va a tener y se puede poner hasta lo que no va a tener. El alcance para este proyecto es el siguiente:

- Envía una solicitud de cotización seleccionando los productos que necesita viéndolos en la sección de productos a los productos y añadiéndolos a un carrito de productos hacia un sistema.
- Muestra la ubicación exacta del local de la empresa computacional Compuexpress.
- Muestra los productos que la empresa tiene para que el cliente pueda ver si hay en stock y si tiene algún descuento.
- Permite compartir con otras personas la aplicación, enviando un mensaje con el link o url al instalador de la aplicación.
- Contacta a los clientes con la empresa computacional vía telefónica y vía mail, para que puedan informarse de cualquier cosa referente a la misma.

- Guarda los datos necesarios, como los productos y cotizaciones, para que la aplicación funcione correctamente en una base de datos.

3. Diseño

En el diseño se realizaron los diseños físicos, lógicos y de pruebas, con sus métricas y el diseño de los cuestionarios. Son esenciales para poder saber cómo programar cada parte de nuestra aplicación.

El diseño lógico nos sirve para poder armar la forma en que va a funcionar nuestro programa, nos ayuda a entender que datos se van a pasar de módulo a módulo, como se va a comportar dicho módulo y que nos va a mostrar como resultado. De esta forma podemos ayudarnos a programar con más facilidad, ya que así logramos entender cada módulo de una manera menos abstracta.

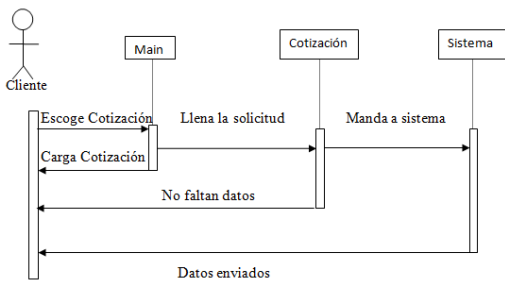


Figura 3.1 Diagrama de secuencias de Cotización

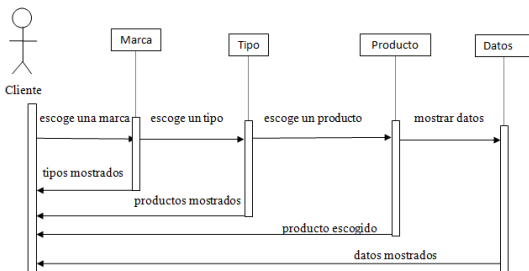


Figura 3.2 Diagrama de secuencias de Productos

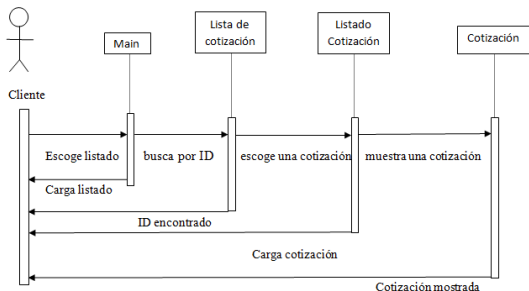


Figura 3.3 Diagrama de secuencias de Lista de cotización

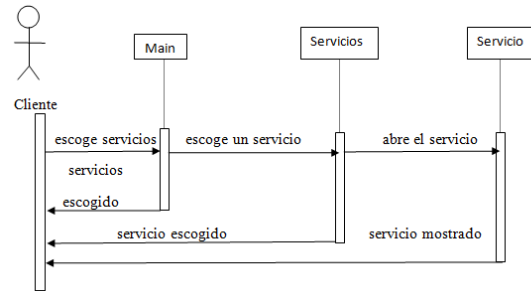


Figura 3.4 Diagrama de secuencias de Servicios

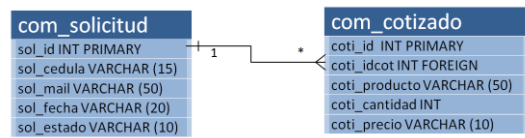


Figura 3.5 Diagrama de entidad relación de solicitud de cotizaciones

Al momento de diseñar cualquier proyecto de software es necesario delimitar la parte física del proyecto, esto es, decidir que módulos van instalados en que equipo, cuales son las configuraciones necesarias para la utilización óptima de nuestro producto y es esto lo que se consigue en el diseño físico.

Existen 3 partes fundamentales en nuestra aplicación:

- Aplicación móvil
- Base de datos
- Unificador



Figura 3.5 Diseño físico computadores separados

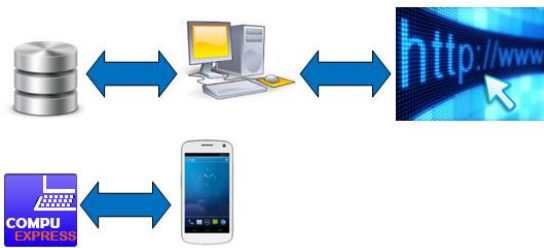


Figura 3.6 Diseño físico un solo computador

El diseño de pruebas se centra en las pruebas que se hacen en el sistema para verificar entradas y salidas esperadas en los requerimientos y validar la interfaz para hacerla amigable con el usuario.

Al momento de calificar hay que ser estricto ya que al encontrar varios errores el usuario dejará de usar nuestra aplicación.

Todo lo que es físico es susceptible a ser medido, sin embargo, un proyecto de software es algo intelectual, algo casi intangible por ese motivo es que muchas de las métricas del software se asemejan. Las métricas se pueden separar en valorar y estimar. Valorar se centra en calidad, fiabilidad y productividad y estimar en valores históricos de esfuerzo y tiempo.

Se realizó cuestionarios, para las pruebas que se han diseñado para la aplicación, así poder anotar las irregularidades que se vayan presentando y así ponderar de alguna forma los resultados. Se decidió utilizar el sistema ROCOF

para usarlo de métrica y lograr tener un ROCOF de 2/100, esto quiere decir que de 100 unidades de tiempo de funcionamiento 2 serán las que se puede aceptar que fallen en envíos de datos.

4. Implementación

Aquí se describen las plataformas que se usaron para poder realizar la aplicación y los problemas que se presentaron al implementar. Se describe la creación de lo que se diseñó en sí.

Como todo proyecto informático es siempre importante el poder establecer que parte será programada en que plataforma. Aquí se divide en dos partes:

- Aplicación móvil.
- Base de datos.

Para la aplicación móvil se decidió usar el IDE Eclipse, Android SDK para la codificación y Adobe Illustrator para la parte visual.

Para la base de datos se decidió usar MySQL Server y el unificador (el que permite conectarlo con la aplicación móvil) se lo hizo en un servidor web.

Ya habiendo pasado por todo el proceso de implementación, después de haber creado la aplicación completa, se describe los problemas que surgieron en el momento de la implementación. Como cualquier proyecto de software, este tuvo continuos problemas.

Como saber qué metáforas usar, limitaciones del emulador, y en donde poner ciertas partes importantes de la aplicación.

5. Pruebas

En las pruebas se realizan todas las pruebas que se diseñaron en el capítulo 3: pruebas de usabilidad, bajo nivel, alto nivel y se realizó el análisis de resultados. También se describen las pruebas de funcionalidad, las cuales son pruebas obligatorias en cualquier proyecto software.

Las pruebas de usabilidad no son más que pruebas que ayudan a los programadores a ver si el producto que están realizando es fácil de usar e intuitivo. Sirven para garantizar calidad y involucramiento de parte de su programa para el cliente.

Las pruebas de usabilidad nos mostraron falencias pequeñas en el mapa, en hacer aparecer la información de la empresa y en las opciones de alejar y acercar, pero en general, se le hizo fácil e intuitivo al usuario la aplicación.

Las pruebas de bajo nivel son sumamente importantes para un proyecto de software. En síntesis lo que permiten estas pruebas es que los involucrados en las interfaces puedan ver si están usando los elementos correctos. Para esto se valen de las interfaces dibujadas a groso modo en papel.

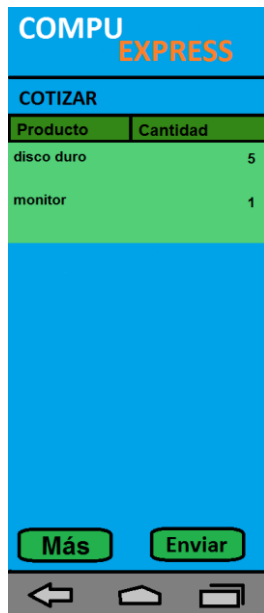


Figura 5.1 Pruebas de bajo nivel

Las pruebas de bajo nivel mostraron varias cosas útiles para mejorar, como el orden en que se presentan las opciones en el menú, al seleccionar una cotización mejorar la metáfora para cancelar la misma.

Las pruebas de alto nivel se asemejan muchísimo a las pruebas de bajo nivel, pero en este caso ya no se hacen las pruebas en dibujos en papel. Estas pruebas se las hace cuando uno ya está a punto de finalizar el proyecto, ya que se las realiza en la computadora o en el dispositivo donde puede ser instalado.

En las pruebas de alto nivel, se logró sacar problemas de metáforas en el botón de pedir cotización en la opción de productos. Pero no se puso como error sino como consejo y observación.

Los resultados de las pruebas de bajo nivel, conjuntamente con las pruebas de usabilidad, nos dieron de forma exitosa una retroalimentación, que logró que las pruebas de alto nivel se hicieran de forma fluida y que sea usado con facilidad la aplicación.

Al momento de haber terminado las pruebas de alto nivel, quedamos satisfechos ya que el producto estaba relativamente perfecto, ya que no se suscitaron errores de interfaz o problemas en el sistema de alguna forma, quedando de resultado que se pueda concluir que el sistema estaba listo para ser entregado al cliente.

Conclusiones y recomendaciones

1. El realizar un proyecto informático, por más pequeño que parezca, abarca una secuencia grande de pasos a seguir como programas necesarios para crearlo, funciones que jamás se pensaba al inicio que se las iba a usar. Y es que, realizando un proyecto de esta magnitud por fuerza propia, se da cuenta que se necesita la ayuda de varias personas para realizar un proyecto, ya que así se demoraría al menos la mitad del tiempo.
2. Los momentos cruciales de todo proyecto son las toma de requerimientos con nuestro cliente o el usuario, los cuales nunca faltan a lo largo del proyecto y en los cuales siempre pedirán cambios imposibles y otros un tanto realizables, aquí es cuando se debe de explicar a nuestro cliente cuales son los

límites de nuestro conocimiento y ofrecerle una solución para que queden satisfechos.

3. Los diseños lógicos y físicos nos muestran el esqueleto de todo nuestro proyecto, sirven para ver cómo debe de estar estructurado y podemos valer de ellos para futuros problemas que se presenten al programar, aunque muy pocos programadores se percaten de tal utilidad.
4. El implementar nuestro proyecto conlleva una investigación constante, para poder solucionar cada problema que no dejan de llegar, son momentos en que se debe tener suma paciencia y buscar momentos de iluminación de cierta forma, ya que a veces son errores de lo más obvios, pero por el estrés se lo ve de un tamaño más grande de lo que realmente es.
5. El último paso del proyecto fueron las pruebas, el cual para muchos programadores es tan tedioso como el documentar el código de los proyectos, aduciendo que eso no sirve de nada, pero ambas cosas son realmente importantes para entregar un producto de calidad a nuestro cliente y se deberían hacer ley en nuestros procesos.

Con todo lo mostrado se logró realizar en su totalidad lo estipulado en los objetivos específicos al igual que obviamente el objetivo general de este proyecto. Todos nuestros objetivos específicos fueron en su medida un reto a elaborar pero se pudo realizar con éxito.

1. Se recomienda a los estimados lectores de este documento que realicen cada uno de los pasos de un proyecto de software sin ningún pero, ya que por más que en el momento pareciera que es una pérdida de tiempo, le ahorrará mucho tiempo y dinero a futuro.
2. Al momento de mostrar un documento que solo es para ser visualizado mas no editado, no complicarse con formas complicadas de visualización. Ver siempre la opción idónea, la cual muchas veces es la opción más sencilla, en este caso usamos `TableLayout`.

Bibliografía

1. *Wikipedia*, Java (lenguaje de programación), Fecha de acceso el 12 de septiembre de 2013,

[http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

2. *Wikipedia*, PHP, Fecha de acceso el 12 de septiembre de 2013, <http://es.wikipedia.org/wiki/PHP>
3. *Wikipedia*, Eclipse (software), Fecha de acceso el 12 de septiembre de 2013, [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))
4. *Wikipedia*, SQL, Fecha de acceso el 12 de septiembre de 2013, <http://es.wikipedia.org/wiki/SQL>
5. *Wikipedia*, SQL Server, Fecha de acceso el 12 de septiembre de 2013, http://es.wikipedia.org/wiki/SQL_Server
6. *Android developer*, Fecha de acceso el 12 de septiembre de 2013, <http://developer.android.com>
7. *Wikipedia*, Caso de uso, Fecha de acceso el 12 de septiembre de 2013, http://es.wikipedia.org/wiki/Caso_de_uso
8. Bernd Bruegge, Allen H. Dutoit. *Object Oriented Software Engineering* (3ra. edición). Estados Unidos: Prentice Hall. (2000). Capítulo 4, pág. 100–106, 118–119.
9. Ian Sommerville. *Ingeniería del software* (7ma. edición). Madrid, España: Pearson. (2005). Capítulo 6, pág. 109-115.
10. Ian Sommerville. *Ingeniería del software* (7ma. edición). Madrid, España: Pearson. (2005). Capítulos 11 - 16, pág. 217-354.
11. *PcWorld*, Android e Ios se disputan el mercado de smartphones este año, Fecha de acceso el 11 de septiembre de 2013, <http://www.pcworld.com.ec/android-e-ios-se-disputan-el-mercado-de-smartphones-este-ano/>