



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
INGENIRÍA EN SISTEMAS COMPUTACIONALES**

TEMA:

**Implementación de un aplicativo web para la requisición,
asesoramiento y cotización de productos para la
construcción de obras civiles y arquitectónicas.**

AUTOR:

Sandoya Mendoza, Manuel Alejandro

**Trabajo de titulación previo a la obtención del título de
INGENIERO EN SISTEMAS COMPUTACIONALES**

TUTOR:

Ing. Pesantes Méndez, Jorge Salvador, Mgs.

Guayaquil, Ecuador

9 de marzo del 2018



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIRÍA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Sandoya Mendoza, Manuel Alejandro**, como requerimiento para la obtención del título de **Ingeniero en Sistemas Computacionales**.

TUTOR

f. _____

Ing. Pesantes Méndez, Jorge Salvador, Mgs.

DIRECTORA DE LA CARRERA

f. _____

Ing. Guerrero Yépez, Beatriz Del Pilar, Mgs.

Guayaquil, a los 9 días del mes de marzo del año 2018



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIRÍA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Sandoya Mendoza, Manuel Alejandro**

DECLARO QUE:

El Trabajo de Titulación, “**Implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas**” previo a la obtención del título de **Ingeniero en Sistemas computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 9 días del mes de marzo del año 2018

EL AUTOR:

f. 
Sandoya Mendoza, Manuel Alejandro



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIRÍA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

AUTORIZACIÓN

Yo, **Sandoya Mendoza, Manuel Alejandro**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **“Implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 9 días del mes de marzo del año 2018

EL AUTOR:

f. 
Sandoya Mendoza, Manuel Alejandro



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIRÍA
INGENIERÍA EN SISTEMAS COMPUTACIONALES**

TRIBUNAL DE SUSTENTACIÓN

f. 

Ing. Guerrero Yépez, Beatriz Del Pilar, Mgs
DIRECTORA DE LA CARRERA

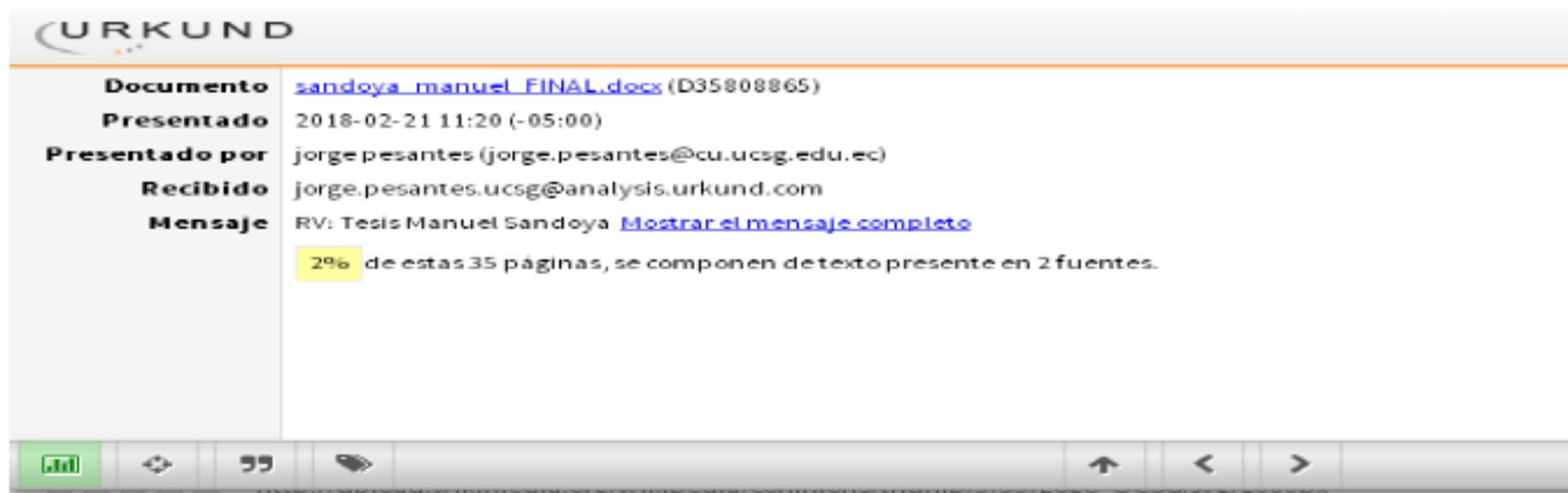
f. 

Ing. Yanza Montalvan, Ángela Olivia
COORDINADORA DEL ÁREA

f. 

Ing. Camacho Coronel, Ana Isabel, Mgs
OPONENTE

REPORTE DE SOFTWARE ANTIPLAGIO



The screenshot displays the URKUND interface with the following details:

- URKUND** logo at the top left.
- Documento:** [sandoya_manuel_FINAL.docx](#) (D35808865)
- Presentado:** 2018-02-21 11:20 (-05:00)
- Presentado por:** jorge pesantes (jorge.pesantes@cu.ucsg.edu.ec)
- Recibido:** jorge.pesantes.ucsg@analysis.orkund.com
- Mensaje:** RV: Tesis Manuel Sandoya [Mostrar el mensaje completo](#)

The main report content states: **2%** de estas 35 páginas, se componen de texto presente en 2 fuentes.

The bottom of the interface features a navigation bar with icons for home, search, and navigation (up, left, right arrows).

AGRADECIMIENTO

A mi padres, Walter y Fátima, mis hermanos, Walter, Miguel, Hernán y Sofía, en los que encontré siempre un apoyo incondicional. A mis maestros, que guiaron cada paso en mi formación profesional. A los ingenieros Jorge y Adela que me acompañaron en el desarrollo de este proyecto.

Sandoya Mendoza, Manuel Alejandro

DEDICATORIA

A mi padre, Arq. Walter Sandoya, a mi querida madre, Fátima Mendoza, magníficos seres humanos que sembraron en mí los valores que hoy forman mi personalidad.

Sandoya Mendoza, Manuel Alejandro

ÍNDICE GENERAL

RESUMEN	XIV
ABSTRACT	XV
INTRODUCCIÓN	16
CAPÍTULO I: EL PROBLEMA	18
1.1 Planteamiento del problema.....	18
1.2 Objetivos.....	18
1.2.1 Objetivo general	19
1.2.2 Objetivos específicos.....	19
1.3 Alcance	19
1.4 Justificación e Importancia.....	20
CAPÍTULO II: MARCO TEÓRICO	21
2.1 Cotizaciones en el ámbito profesional de la construcción	21
2.2 La web 2.0 o web social y el tratamiento interactivo de la información.....	22
2.3 Conceptualizaciones técnicas	23
2.4 Base legal.....	30
CAPÍTULO III: METODOLOGÍA DE LA INVESTIGACIÓN	32
3.1 Tipo y método de investigación.....	32
3.1.2 Población, muestra y muestreo.....	36
3.1.3 Técnicas e instrumento para la recolección de información.....	36
3.1.4 Herramienta para procesamiento de información	37
3.2 Análisis de resultados.....	37
3.2.1 Encuesta a profesiones de la construcción.....	37
3.2.2 Encuesta a proveedores de productos para la construcción	42
CAPÍTULO IV: PROPUESTA TECNOLÓGICA	47
4.1 Herramientas de desarrollo	47
4.1.1 Visual Studio Code.....	47

4.1.2	Bash de Ubuntu en Windows 10.....	47
4.1.3	MongoDB Community Edition	48
4.2	Lenguaje de programación y <i>framework</i>	49
4.3	Estructura y organización de la aplicación.....	51
4.3.1	La carpeta <i>app</i>	52
4.3.2	La carpeta <i>config</i>	53
4.3.3	El archivo <i>Gemfile</i>	55
4.4	Modelado de la base de datos	57
4.4.1	Colección Usuarios	57
4.4.2	Colección Requirentes	58
4.4.3	Colección Proveedores	58
4.4.4	Colección Requisición	60
4.4.5	Colección Productos	60
4.4.6	Colección Cotizaciones y colección Elementos	60
4.4.7	Colección Categorías	60
4.4.8	Colección Comentarios	61
4.5	Políticas de autorización	64
4.5.1	Políticas de requisiciones.....	64
4.5.2	Políticas de cotizaciones.....	64
4.5.3	Políticas de productos	64
4.6	Secciones de la aplicación.....	65
4.6.1	Manejo de sesiones	65
4.6.2	Registro de usuarios	65
4.6.3	Requisiciones.....	65
4.6.4	Ingreso de productos	66
4.6.5	Cotizaciones.....	66
4.6.6	Modificación de registros	66
4.6.7	Comentarios.....	66

4.6.8	Panel de control.....	67
4.7	Servicios externos.....	71
4.7.1	Servicio de mensajería Gmail	71
4.7.2	Almacenamiento de imágenes en <i>bucket</i> S3 de AWS.....	72
4.7.3	Plataforma como servicio (PaaS) Heroku	73
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....		75
5.1	Conclusiones.....	75
5.2	Recomendaciones	76
BIBLIOGRAFÍA.....		77
ANEXOS.....		79

ÍNDICE DE FIGURAS

Figura 2.1 Vista general del modelo cliente / servidor.....	24
Figura 2.2 Lenguajes del lado del cliente.	25
Figura 2.3 Comparación entre las mismas instrucciones de código escrito en JavaScript sin librerías y código escrito utilizando la librería jQuery.....	26
Figura 2.4 Modelo vista controlador	29
Figura 2.5 Flujo de datos de MongoDB al cliente.....	30
Figura 3.1: Fases del proceso cuantitativo.	33
Figura 3.2: Proceso y roles de Scrum.....	35
Figura 3.3: Pregunta 1 y gráfico. Encuesta a profesionales: medios de información	38
Figura 3.4: Pregunta 2 y gráfico. Encuesta a profesionales: utilización de internet.....	38
Figura 3.5: Pregunta 3 y gráfico. Encuesta a profesionales: costos de productos	39
Figura 3.6: Pregunta 4 y gráfico. Encuesta a profesionales: nivel de dificultad.	40
Figura 3.7: Pregunta 5 y gráfico. Encuesta a profesionales: fuentes de información.	40
Figura 3.8: Pregunta 6 y gráfico. Encuesta a profesionales: fuentes de información.	41
Figura 3.9: Pregunta 7 y gráfico. Encuesta a profesionales: aplicación web.	42
Figura 3.10: Pregunta 1 y gráfico. Encuesta a proveedores: medios de comunicación.....	42
Figura 3.11: Pregunta 2 y gráfico. Encuesta a proveedores: internet.	43
Figura 3.12: Pregunta 3 y gráfico. Encuesta a proveedores: clientes potenciales.....	44
Figura 3.13: Razones de la pregunta 3.....	44
Figura 3.14: Pregunta 4 y gráfico. Encuesta a proveedores: cotización de productos.	45
Figura 3.15: Pregunta 5 y gráfico. Encuesta a proveedores: aceptación de aplicativo.....	45

Figura 3.16: Pregunta 6 y gráfico. Encuesta a proveedores: requerimiento en aplicación.....	46
Figura 4.1: Consola de comando Bash de Ubuntu en Windows 10	48
Figura 4.2: Entorno de desarrollo Visual Studio Code	49
Figura 4.3: Utilización del framework Ruby on Rails en comparación con sus similares en los últimos cinco años.....	50
Figura 4.4: Estructura de la aplicación cotipro autogenerada por Ruby on Rails.....	51
Figura 4.5: Carpeta app dentro de la estructura de Cotipro	53
Figura 4.6: Carpeta config dentro de la estructura de Cotipro	53
Figura 4.7: Archivo de configuración de rutas routes.rb.....	54
Figura 4.8: Ejecución del comando rails routes que permite ver las rutas configuradas en routes.rb. En la figura no se muestran todas las rutas.	54
Figura 4.9: Resumen del archivo mongoid.yml, que establece las conexión a la base de datos en los ambiente de desarrollo, prueba y producción.	55
Figura 4.10: Primeras líneas del archivo Gemfile. A partir de la línea 10 la definición de las gemas.	56
Figura 4.11: Ejecución del comando bundle, que descarga e instala las gemas declaradas en el Gemfile desde rubygems.org.....	56
Figura 4.12: Gemas agregadas al proyecto, en el archivo Gemfile	57
Figura 4.13: Colección de usuarios definida en el modelo User	58
Figura 4.14: Colección Requirentes definida en el modelo Requester	59
Figura 4.15: Colección Proveedores definida en el modelo Provider.....	59
Figura 4.16: Colección Requisición definida en el modelo Requisition	61
Figura 4.17: Colección Productos definida en el modelo Products	62
Figura 4.18: Colección Cotizaciones definida en el modelo Quotation	62
Figura 4.19: Colección Elementos de cotización definida en el modelo Item.....	63
Figura 4.20: Colección Categorías definida en el modelo RequisitionTag...63	63
Figura 4.21: Colección Comentarios definida en el modelo Comment	63
Figura 4.22: Control de sesiones en Cotipro.....	67
Figura 4.23: Formularios de registros de requirentes y proveedores, en orden de izquierda a derecha	68
Figura 4.24: Formulario de registro de requisiciones.	68

Figura 4.25: Formulario de registro de productos.	69
Figura 4.26: Formulario de registro de cotizaciones.	69
Figura 4.27: Ejemplo de modificación de registro. Modificando requisición.	70
Figura 4.28: Sección de comentarios en una cotización.....	70
Figura 4.29: Panel de control del usuario Administrador.	71
Figura 4.30: Correo de verificación de cuenta de Cotipro.....	71
Figura 4.31: Correo de notificación de cotización realizada.....	72
Figura 4.32: Bucket 'cotipro' en S3 de AWS.....	72
Figura 4.33: Parte del dashboard de Heroku, en donde se encuentra alojada la aplicación	74

RESUMEN

El presente proyecto de titulación consiste en el análisis de las formas en cómo profesionales y proveedores en el ámbito de la construcción se desenvuelven en el proceso de realizar u obtener una cotización, y cómo la implementación de un aplicativo en la web, basado en los principios de la web 2.0, puede mejorar el mismo. El análisis se enfoca en las actuales limitaciones que tanto profesionales y proveedores de la construcción tienen al momento de conseguir información sobre un producto o encontrar clientes potenciales, según corresponda. Así mismo, en el presente documento se explica cómo se desarrolló el aplicativo web que pretende mejorar este proceso, detallando las diferentes tecnologías que se utilizaron para alcanzar los objetivos planteados. Por último, se muestran las conclusiones alcanzadas en este proyecto y se dan recomendaciones para realizar iteraciones de mejoras en el aplicativo web propuesto.

Palabras clave: APLICATIVO WEB; REQUISICIÓN; COTIZACIÓN; CONSTRUCCIÓN; PROVEEDORES

ABSTRACT

The present titling project consists of the analysis of the ways in which professionals and suppliers in the field of construction are developed in the process of making or obtaining a quote, and how the implementation of an application on the web, based on the principles of the web 2.0, you can improve it. The analysis focuses on the current limitations that both professionals and construction providers have when they get information about a product or find potential customers, as appropriate. Likewise, this document explains how the web application was developed that aims to improve this process, detailing the different technologies that were used to achieve the objectives set. Finally, the conclusions reached in this project are shown and recommendations are given to carry out iterations of improvements in the proposed web application.

Keywords: WEB APPLICATION; REQUISITION; QUOTATION; CONSTRUCTION; SUPPLIERS

INTRODUCCIÓN

El intercambio de información, conocimientos y experiencias es un hábito natural en el ámbito profesional y es constante durante la realización de cualquier proyecto. Desde hace algunas décadas, el tener acceso a internet garantiza una amplia gama de información la cual puede filtrarse bajo parámetros o requerimientos propios a través de los grandes motores de búsqueda que existen. Sin embargo, mantenerse actualizado y saber quién o quiénes pueden proporcionar la información adecuada a los requerimientos de las personas, es algo que demanda de un largo proceso de búsqueda, ubicación e inscripción en los canales de información en donde la solicitada normalmente se desenvuelve.

Un canal de información en internet es lo que se define dentro del concepto de la web 2.0 o web social: un espacio virtual de intercambio de información, como pueden ser wikis, blogs o redes sociales, en donde sus integrantes generan contenido relevante a las temáticas a las que están suscritos. Estos canales son aprovechados por muchos profesionales de distintas áreas para actualizar sus conocimientos e intercambiar experiencias.

Por otro lado, estos espacios virtuales no quedan limitados a que sus usuarios puedan intercambiar información, también es posible llegar a realizar intercambios o compras de productos y servicios fuera de la web tras llegar a un mutuo acuerdo. Esto es posible gracias al avance de las herramientas utilizadas para la interacción social en internet.

Justamente el intercambio de productos a través de la web es realmente útil para muchos profesionales que requieren de una vía rápida y eficaz para conseguir los materiales o productos con los que trabajan. Sin embargo, en algunas áreas profesionales, no es habitual recurrir a los espacios virtuales de internet para realizar consultas sobre productos ya que generalmente están supeditados a consultar más a gente cercana con experiencia o directamente a proveedores conocidos, esto debido a que no existen o son muy pocos los espacios para realizar esta búsqueda en internet, además de ser en ciertos casos ineficiente.

Entre los profesionales a los que no le es del todo eficiente la tarea de encontrar y obtener información sobre productos en internet, y en los que se enfoca el presente trabajo de titulación, están lo que se dedican al área de la construcción, que se enfrentan a una larga búsqueda y recolección de información antes de obtener los datos suficientes con los que pueden realizar cotizaciones de los productos que necesitan o poder plantearse alternativas.

Se debe agregar que, esta búsqueda siempre queda limitada a los conocimientos previos que tenga el profesional sobre dónde encontrar determinada información. Por otra parte, los proveedores de productos de la construcción también quedan limitados a que en internet sean sus clientes potenciales quienes los busquen a ellos y no al contrario.

Tomando en cuenta estos problemas, se propone la creación de un aplicativo alojado internet, siguiendo el concepto de la web 2.0, en donde profesionales de la construcción y proveedores puedan converger e intercambiar información sobre costos y calidad de productos además de experiencias en ámbitos constructivos.

Los resultados de esta investigación se encuentran organizados en este documento de la siguiente forma: en el capítulo I se explica problemática a resolver, así como los objetivos, alcances y justificativos de este proyecto; en el capítulo II se revisa el marco teórico, las conceptualizaciones técnicas y la base legal que sustentan el desarrollo del aplicativo web propuesto; en el capítulo III se detalla el marco metodológico de esta investigación y el análisis de los datos obtenidos durante el levantamiento de información; en el capítulo IV se presenta la propuesta tecnológica que permitió resolver el problema planteado; en el capítulo V se dan las conclusiones y recomendaciones alcanzadas en este proyecto. Finalmente, el documento termina con las referencias bibliográficas y los anexos respectivos.

CAPÍTULO I: EL PROBLEMA

Para abordar el tema relacionado con el manejo e intercambio de información relativo a materiales de construcción entre profesionales y proveedores, es necesario iniciar un análisis de la problemática presentada, con el fin de establecer las formas de diseñar posibles soluciones.

En este capítulo se hace una revisión global del punto de inicio de la investigación, pasando por los objetivos y su justificación.

1.1 Planteamiento del problema

Para encontrar información sobre un producto para la construcción de obras civiles o arquitectónicas en internet, los profesionales en esta área se enfrentan a largos procesos de búsqueda y registro manuales de esa información para poder elaborar sus cotizaciones, además, en la mayoría de los casos, están limitados a fuentes previamente conocidas, lo que restringe la posibilidad de obtener alternativas o mejores productos en el mercado.

Por otro lado, los proveedores de productos de esta categoría, que están en constante promoción de sus productos a través de internet o medios tradicionales de publicidad, no consiguen muchas veces llegar a sus clientes potenciales, dado que resulta difícil (y que depende de costosos estudios de mercado) encontrar la mayor demanda de los productos que se ofertan.

Por lo tanto, la existencia de un aplicativo web, que permita el contacto directo de profesionales de la construcción y sus proveedores, puede ser beneficioso para ambos, ya que permitiría una oferta directa por parte de los proveedores y una obtención rápida de información de los productos que los profesionales de esta área necesitan para sus obras.

1.2 Objetivos

Con la intención de solucionar la problemática planteada, se definieron los siguientes objetivos:

1.2.1 Objetivo general

Implementar un aplicativo web que facilite a los profesionales de la construcción conseguir asesoramiento sobre la calidad y costos de productos y materiales para la construcción de distintos proveedores.

1.2.2 Objetivos específicos

- Determinar las formas en la que los profesionales de la construcción encuentran información sobre productos, con el fin de establecer las necesidades de rediseño de las mismas.
- Diseñar y desarrollar un sistema que facilite el intercambio eficaz de información, conocimiento y experiencias de productos para la construcción.
- Implementar un aplicativo web para la distribuidora Gary Giler que permita a los profesionales de la construcción consultar información sobre productos.

1.3 Alcance

Este trabajo de investigación está orientado al diseño de un aplicativo web que ofrece un sistema de recepción de requerimientos de profesionales de la construcción sobre productos en los que están en búsqueda y necesitan para sus respectivas obras civiles o arquitectónicas. Los requerimientos que son procesados por este sistema son ingresados y publicados con el objetivo de que los proveedores de estos productos sean notificados de estas demandas del mercado.

La información que es entregada por los proveedores hacia sus clientes, que puede ir desde precios y ofertas (en una cotización) hasta asesoramiento personalizado (como comentario adicional a la cotización), es seleccionada o descartada por el requirente (profesional de la construcción), el cual finalmente se pone en contacto con el proveedor para realizar una transacción (por fuera de la aplicación). Esta vía de comunicación pretende ampliar la visión del mercado de los profesionales de la construcción y mejorar las ofertas de los proveedores.

El aplicativo contará con secciones en donde intervendrán usuarios clientes (promotores y profesionales de la construcción) y usuarios proveedores, los primeros, con el rol de poder realizar requisiciones de productos, y los segundos, con el rol de poder ofertar. Además, un usuario administrador y moderador de los contenidos publicados.

Tanto los usuarios clientes como los usuarios proveedores contarán con una sección dentro de la cotización en donde podrán intercambiar comentarios acerca de la misma, esto permitirá aclarar o complementar la información necesaria antes de que se dé la comunicación directa (por fuera de la aplicación).

La implementación estará alojada en la plataforma como servicio *Heroku* y la base de datos será *mongoDB*. Las imágenes y archivos subidos al aplicativo web, como muestras visuales de un producto, serán almacenados en un repositorio S3 de Amazon Web Services (AWS).

1.4 Justificación e Importancia

El diseño de este aplicativo web responde a la necesidad de mejorar los canales de información en internet de los profesionales de la construcción y optimizar los procesos de oferta de los proveedores de materiales, que actualmente se desenvuelven en canales ineficientes para ofrecer sus productos.

La experiencia resultante del desarrollo y utilización de este aplicativo web podría ser replicado en otros campos profesionales en donde se requiera de constante intercambio de información y actualización de experiencias para la compra de productos.

Además, este proyecto de titulación responde a la línea de investigación de la carrera de Ingeniería en Sistemas Computacionales de la Universidad Católica de Santiago de Guayaquil denominada: "investigación y desarrollo de nuevos servicios o productos".

CAPÍTULO II: MARCO TEÓRICO

Para comprender los conceptos que sustentan el presente trabajo de titulación, en este capítulo son revisados y analizados el entorno en el que se desenvuelve la aplicación propuesta, así como también las tecnologías, métodos y paradigmas que permitieron el desarrollo de la misma, además de sus bases legales.

2.1 Cotizaciones en el ámbito profesional de la construcción

En el ámbito profesional de la construcción, realizar cotizaciones de materiales o productos para la construcción es una tarea importante y significativa en cualquier proyecto de edificación, reparación o remodelación. A lo largo de este documento son manejados conceptos que son definidos dentro del aplicativo web propuesto, y que intentan reflejar a los sujetos involucrados en la realización de una cotización, es decir, el requirente y proveedor de la información de los materiales o productos.

En cuanto a los objetivos y tareas que los sujetos involucrados en la cotización realizan, se tiene que ambos realizan una búsqueda constante del otro con el fin de obtener la información que necesitan, siendo una tarea primordial que en este documento se define como “búsqueda de fuente”, y está sujeta siempre a las circunstancias y contactos del sujeto, es decir, si tiene o no los recursos y conocimientos para realizar dicha búsqueda.

Así mismo, los sujetos realizan otras tareas una vez encontrada su fuente. Estas son, en cuanto al proveedor, la realización de ofertas con respecto a la demanda del requirente encontrado y la realización de una cotización para proponer una oferta definitiva; el requirente, en cambio, analiza las ofertas y cotizaciones del proveedor encontrado, aceptando o no cada una de ellas. Dentro del aplicativo web propuesto, lo anteriormente mencionado se denomina como una “operación de cotización”.

Se debe agregar que, las cotizaciones en el presente proyecto de titulación, son tratadas como una simple lista informativa de precios para el requirente y una herramienta de realización de ofertas para el proveedor.

2.2 La web 2.0 o web social y el tratamiento interactivo de la información

La característica actual y más importante en la *World Wide Web* es el poder interactuar con la información que fluye a través de las páginas web. Esta característica, que lleva a las páginas web a ser llamadas por muchos más como aplicaciones web, es el resultado evolutivo de una web que antes solo permitía recibir y leer información, y es comúnmente conocida como web 2.0 o web “social”.

Como señala Zofío Jiménez (2013, p. 9) “la web 2.0 es más una actitud que una tecnología”, dado que se trata de que los creadores de web generen aplicaciones web capaces de imitar lo que hace una aplicación de escritorio conectada a una base de datos local, es decir, generar, almacenar y compartir contenido. Un ejemplo de esto son los foros, blogs, y más representativamente, las redes sociales.

Jiménez también menciona algunas de las características principales las aplicaciones web 2.0:

- La Web es la plataforma.
- La información es el procesador.
- Los efectos de la Red son movidos por una arquitectura de participación.
- La innovación surge de características distribuidas por desarrolladores independientes.
- El fin del círculo de adopción de software.

El concepto que es más aplicado al momento del desarrollo de una aplicación web 2.0, es la de que esta sea un servicio. Un servicio web comprende “una serie de tecnologías, protocolos y estándares que permiten el diálogo entre sistemas informáticos. Independientemente de su plataforma, utilizan los propios recursos de comunicación que ofrece Internet” (Zofío Jiménez, 2013, p. 11).

Debe entenderse que los conceptos y términos de la Web 2.0 antes explicados en esta sección, son de suma importancia para comprender

aplicación web propuesta en el presente trabajo de titulación, que funciona como un servicio en el que el tratamiento de la información es dado entre profesionales de la construcción y proveedores de productos.

2.3 Conceptualizaciones técnicas

A lo largo de este documento son manejados términos que tienen que ver con las implicaciones técnicas que permiten el desarrollo de una aplicación web, desde el modelo arquitectónico, pasando por los lenguajes de programación, hasta los ambientes donde la aplicación es ejecutada en producción, es decir, publicada y distribuida al público general.

En primer lugar, se debe entender que el desarrollo de aplicaciones web comprende un variado uso de tecnologías, y que estas son determinadas según el tipo de requerimientos que se necesiten para que estas cumplan el objetivo de la aplicación. En la definición de las tecnologías a utilizar, se tiene en claro que existen dos lados a los que cubrir en el desarrollo web: el lado del cliente y el lado del servidor, el primero, ejecutado generalmente en un navegador de internet y el segundo, el que permite la interacción entre una base de datos y el cliente.

Al hablar de estos dos lados del desarrollo web, que se puede referir en otras palabras como el modelo cliente / servidor (figura 2.1), se habla de una configuración de arquitectura determinada, que en este caso es la más habitual, utilizada y recomendada. Este modelo está “basado en la idea de servicio, en el que el cliente es un componente consumidor de servicios y el servidor es un proceso proveedor de servicios” (Vara Mesa et al., 2014, p. 14).

En ambos lados del desarrollo web se utilizan lenguajes de programación, pero en la mayoría de los casos solo es necesario evaluar y definir el o los lenguajes que se utilizarán del lado del servidor, en donde existen variadas opciones. Desde el lado del cliente, el lenguaje de programación predeterminado es JavaScript (actualmente este puede estar presente también del lado del servidor), el cual es interpretado por todos los navegadores de internet desde las primeras versiones de los mismos hasta la actualidad.

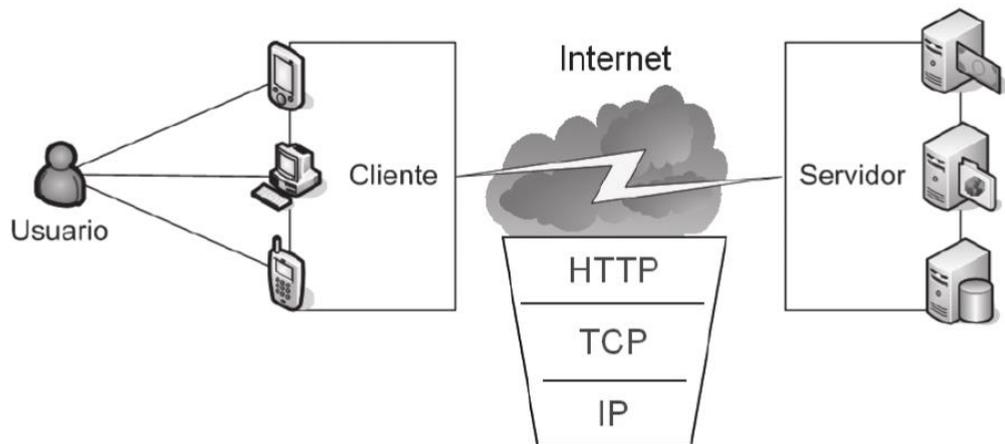


Figura 2.1 Vista general del modelo cliente / servidor. Tomado de "Desarrollo web en entorno servidor", por López Sanz et al., 2014, Madrid: RA-MA Editorial

Se debe agregar que, aunque JavaScript sea el lenguaje de programación del lado del cliente, no es el único lenguaje como tal, existen otros lenguajes que no son de programación pero que sirven para darle estructura y diseño a la vista de cualquier aplicación web, los más importantes y utilizados son: el lenguaje de marcado HTML (*Hypertext Markup Language*) y el lenguaje de estilos CSS (*Cascading Style Sheet*), sin al menos estos dos, no existe una web visual como tal para el usuario, es decir, la interfaz web como se la conoce hoy en día.

Para comprender un poco más las tecnologías utilizadas del lado del cliente, se debe tener en claro sus conceptos más básicos. Comenzando con los lenguajes de marcado y estilos, tenemos que un lenguaje de marcado es "un lenguaje que anota el texto de modo que el ordenador puede manipularlo" (Barzanallana Asensio, 2012), entendiéndose como anotaciones a: identificadores (marcas) preestablecidos que sirven como instrucciones para el computador (y más específicamente, en el caso del HTML, el navegador) para mostrar texto de una manera determinada, y por otro lado, los lenguajes de estilos permiten definir un conjunto de reglas en uno o varios documentos de texto plano que sirven para indicar a un navegador web las modificaciones gráficas o de diseño que debe presentar un documento escrito con un lenguaje de marcado.

En el caso del lenguaje de programación JavaScript, utilizado en el lado del cliente, permite que una página web se convierta en una aplicación

dinámica (también llamada web dinámica). Otra consideración importante es la que señala Mohedano, Saiz, & Salazar Román (2012):

Desde el punto de vista técnico JavaScript es un lenguaje interpretado, eso significa que las instrucciones son analizadas en secuencia por el intérprete de JavaScript del navegador Web, de manera que su ejecución es inmediata a la interpretación. Esto permite que, una vez escrito un programa en JavaScript con un editor de texto plano y embebido el código en un fichero HTML, el navegador es capaz de interpretarlo y ejecutarlo sin necesidad de procesos intermedios. (p. 9)



Figura 2.2 Lenguajes del lado del cliente. *Elaboración propia.*

En la actualidad, JavaScript no es utilizado por sí solo para dar interactividad a las páginas web, existe una gran variedad de librerías (componentes de código reutilizables) que agregan características al lenguaje, que abstraen la complejidad lógica de determinados algoritmos o que sirven de base para desarrollar experiencias de usuario avanzadas de forma sencilla. La más popular y utilizada de estas librerías es jQuery, porque “permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web”. (Córcoles Tendero & Montero Simarro, 2014, p. 134)

Del párrafo anterior cabe aclarar que el DOM (*Document Object Model*) se refiere a la estructura de marcas de un documento HTML y que AJAX (*Asynchronous JavaScript And XML*) es la forma de interactuar con el lado del servidor, lo que permite que el modelo Cliente / Servidor funcione.

Por otra parte, la gran aceptación que tiene la librería jQuery se debe a que, además de ser de código abierto y libre distribución (software libre bajo licencia MIT y GNU v2), permite al momento del desarrollo escribir menos código comparado con usar JavaScript directamente.

JavaScript	
1	<code>var g = document.getElementsByClassName('goodbye');</code>
2	<code>var i;</code>
3	<code>for (i = 0; i < g.length; i++) {</code>
4	<code> g[i].className = g[i].className + 'selected';</code>
5	<code>}</code>
JavaScript + jQuery	
8	<code>\$('.goodbye').addClass('selected');</code>

Figura 2.3 Comparación entre las mismas instrucciones de código escrito en JavaScript sin librerías y código escrito utilizando la librería jQuery.

En el desarrollo del lado del servidor son analizados varios aspectos, como el o los lenguajes de programación que deben utilizarse, el si es necesario utilizar algún esquema de desarrollo predefinido (*framework*) y el motor de base de datos que definirá la organización de la información que ingrese a la aplicación. Toda definición de las tecnologías a utilizar está limitada a las consideraciones que el o los desarrolladores crean pertinente, y en el presente documento se analizan y se ahonda en los conceptos de las tecnologías utilizadas en la aplicación propuesta.

Tomando en cuenta que actualmente los lenguajes que utilizan el paradigma de la programación orientada a objetos son los más recomendados y utilizados en el desarrollo web (y en el desarrollo de software en general), en la aplicación propuesta se optó por la selección de un lenguaje de programación con esta característica.

Tal y como señala Noguera Otero & Riera Terrén (2013), el paradigma de programación orientado a objetos se define como:

un programa que está formado por un conjunto de objetos independientes que interactúan entre ellos. Cada objeto contiene su propio código y sus datos, tiene la capacidad de recibir peticiones

(mensajes de otros objetos) para hacer alguna tarea, procesar datos y enviar peticiones (mensajes) a otros objetos. (p. 125)

En el campo del desarrollo web existen varias opciones de lenguajes de programación orientados a objetos (POO) que pueden utilizarse. Si se toma en cuenta las estadísticas de Github, la plataforma web de desarrollo colaborativo más importante de la comunidad de software, se tiene que los cinco lenguajes de POO más utilizados y que tienen un gran soporte actualmente son:

1. JavaScript (Node.js),
2. Ruby,
3. PHP,
4. Python, y
5. C#.

Cualquiera de los lenguajes antes mencionados puede cubrir el desarrollo del lado del servidor de la aplicación propuesta, por lo que la definición del lenguaje finalmente quedó limitada, como antes se explicaba, a los criterios propios del autor.

Considerando la versatilidad y rápida curva de aprendizaje del lenguaje de programación Ruby, se optó por este para la aplicación propuesta. Ruby se define asimismo como un lenguaje de programación interpretado, reflexivo y orientado a objetos, que es creado por el japonés Yukihiro “Matz” Matsumoto y publicado en 1995 (Muñoz Eslava, 2012, p. 188).

Una vez elegido el lenguaje de programación, se debe decidir si utilizar algún esquema de desarrollo predefinido, es decir, un *framework*. Un *framework*, en el ámbito del desarrollo web (y del software en general), puede definirse como una aplicación configurable e incompleta, estructurada con componentes o módulos de software concretos, que sirven de base para el desarrollo y organización de la arquitectura de una aplicación. Una de las ventajas de los *frameworks* es que lleva a los desarrolladores a realizar buenas prácticas estructuración y codificación.

Dado que, en el ambiente de *Ruby*, el *framework* escrito en este lenguaje que es altamente soportado por su comunidad, además de ser el más utilizado y recomendado para el desarrollo web, es *Ruby on Rails* (o simplemente *Rails*), se optó por agregar a este a la arquitectura de la aplicación propuesta como parte su estructura base. Como todo *framework*, *Rails* permite el desarrollo ágil y efectivo de aplicaciones web, siendo su característica diferenciadora y más popular la amplia abstracción de la complejidad en el manejo y configuración de la base de datos, lo permite una disminución significativa de tiempo en el desarrollo.

A *Ruby on Rails* también se lo puede definir como una librería, que en el entorno de *Ruby* son llamadas *gems* (gemas). A su vez, *Ruby on Rails* como gema es una colección de varias otras gemas que proveen, como componentes de software, la estructura principal del *framework*. Justamente el manejo de gemas dentro del *framework* es una de sus características relevantes, ya que la adición de nuevos requerimientos o módulos completos en la aplicación puede ser provista a través de una gema desarrollada por autores externos.

Otra característica relevante de *Ruby on Rails* es la utilización del modelo vista controlador (MVC), que como describe Granados La Paz (2014) es:

Un patrón de diseño, basado en tres componentes claramente diferenciados: modelo: contiene los datos de la aplicación, implementando la lógica y funciones; vista: representa la información contenida en el modelo; y controlador: actúa de intermediario, recibiendo la interacción del usuario a través de la vista y manipulando el modelo. (p. 331)

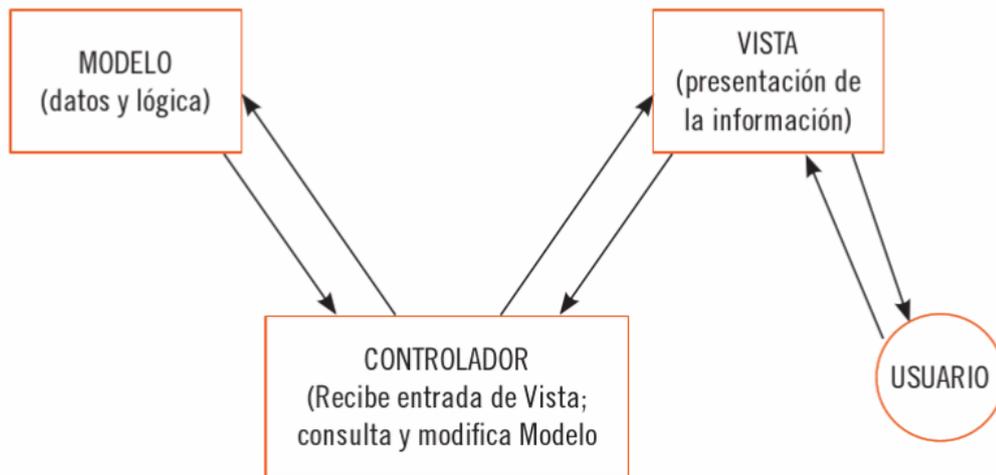


Figura 2.4 Modelo vista controlador. Tomado de "Desarrollo de aplicaciones web en el entorno servidor ", por Granados La Paz, 2014, Madrid: IC Editorial

Para complementar los conceptos en el lado del servidor, se debe definir las cuestiones referentes a la base de datos. Específicamente, en la aplicación propuesta, la base de datos es del tipo NoSQL (*Not only SQL*), por lo que en el presente documento se hace referencia solamente a este tipo de base de datos, que se diferencian de las bases tradicionales que utilizan el paradigma relacional.

Las bases de datos NoSQL ofrecen numerosas ventajas, una de ellas, como señala Sarasa (2016, p. 24): “proporcionan un modelo de datos que encaja mejor con las necesidades de las aplicaciones simplificando así la interacción, lo que resulta en tener que codificar, depurar y evolucionar menos las aplicaciones”. También destaca, como principal característica, la no existencia de un esquema, “lo que permite añadir libremente campos a los registros de la base de datos sin tener que redefinir la estructura”. (p. 25)

Existen distintos modelos bases de datos NoSQL, para la aplicación propuesta se optó por el modelo orientado a documento, en el que se destaca MongoDB, que es la base de datos utilizada. Los modelos de datos documental, explica Sarasa (2016, p. 45) “se caracterizan porque definen un conjunto de estructuras y tipos permitidos que pueden ser almacenados, y además es posible acceder a la estructura del agregado, teniendo como ventaja que se consigue más flexibilidad en el acceso”.

Aterrizando en su definición, MongoDB es finalmente una base de datos NoSQL documental, que almacena los datos de forma no esquematizada en documentos siguiendo un formato basado en JSON (*JavaScript Object Notation*), denominado BSON (*Binary JSON*). Esta base de datos fue creada por 10gen en el año 2007.

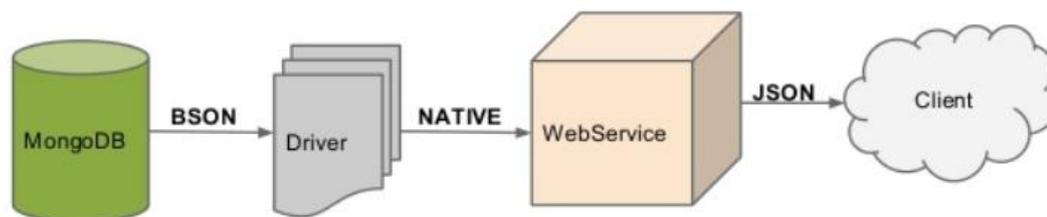


Figura 2.5 Flujo de datos de MongoDB al cliente. Tomado de “BSON Mad Science for fun and profit”, por Alessandro Molina, 2013

2.4 Base legal

En el marco de la Ley de comercio electrónico, firmas electrónicas y mensajes de datos (ley no. 2002-67), se establecen que los mensajes de datos, que en el caso de la aplicación propuesta son todas las cotizaciones realizadas por proveedores, tienen validez jurídica al igual que los documentos escritos, según el artículo 2 de la ley antes mencionada.

Otros artículos que garantizan la validez de las cotizaciones son: Art. 8, Art. 9 y Art. 10, que revisan la conservación, protección y procedencia del mensaje de datos respectivamente. También, tanto el proveedor como el requirente están sometidos a los artículos 48 a 50 sobre sus derechos como consumidores de los servicios que ofrece la aplicación.

Por otra parte, en el marco del Plan Nacional del Buen Vivir, que contempla las “Tecnologías, innovación y conocimiento” como parte de sus estrategias, en su objetivo 10 para “Impulsar la transformación de la matriz productiva”, se describe en su punto 10.3 de “Diversificar y generar mayor valor agregado en los sectores prioritarios que proveen servicios”, una política que comprende el apoyo y fortalecimiento a la industria local de servicios, y en las que se beneficia, por tanto, el desarrollo de una industria de software

libre, que favorece significativamente el desarrollo y avances técnicos en el país.

Es así que, y apoyando los esfuerzos gubernamentales para el cambio de la matriz productiva, la aplicación propuesta se presenta como un desarrollo de software en el que su código es de libre distribución para su utilización en investigación o integración en otras soluciones.

CAPÍTULO III: METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se describe el tipo y método de investigación utilizado en el presente proyecto de titulación. Así mismo, se describirá la población a la que está enfocado esta investigación y la muestra tomada, así como la forma en cómo se recolectó los datos y el análisis de los mismos.

3.1 Tipo y método de investigación

El presente trabajo de titulación está orientado en el tipo de investigación descriptiva que, tal como describen Hernández Sampieri, Fernández Collado, & Baptista Lucio (2014, p. 92), “se busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis”, lo que implica para esta investigación, el levantamiento de información sobre profesionales de la construcción y proveedores, específicamente en sus tareas de búsqueda y cotización de productos para la construcción, respectivamente.

Esta investigación también tiene un enfoque metodológico cuantitativo, ya que se recolectan datos sobre el comportamiento de los sujetos en estudio (profesionales y proveedores de la construcción) y se realiza un análisis estadístico para probar la hipótesis planteada, es decir, comprobar la factibilidad de la aplicación web propuesto.

Según Hernández Sampieri et al. (2014, p. 4), la característica principal que diferencia al enfoque metodológico cuantitativo es que se define como un conjunto de procesos secuenciales y probatorios, es decir, “cada etapa precede a la siguiente y no podemos ‘brincar’ o eludir pasos”. También explica que:

Parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se traza un plan para probarlas (diseño); se miden las variables en un determinado contexto;

se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto de la o las hipótesis.

Como se refleja en la figura 3.1, esta metodología se compone de 10 fases plenamente definidas. Para el caso de la presente investigación, estas 10 fases se siguieron para obtener los resultados que darían la información necesaria para levantar los requerimientos funcionales y no funcionales de la aplicación web propuesta.

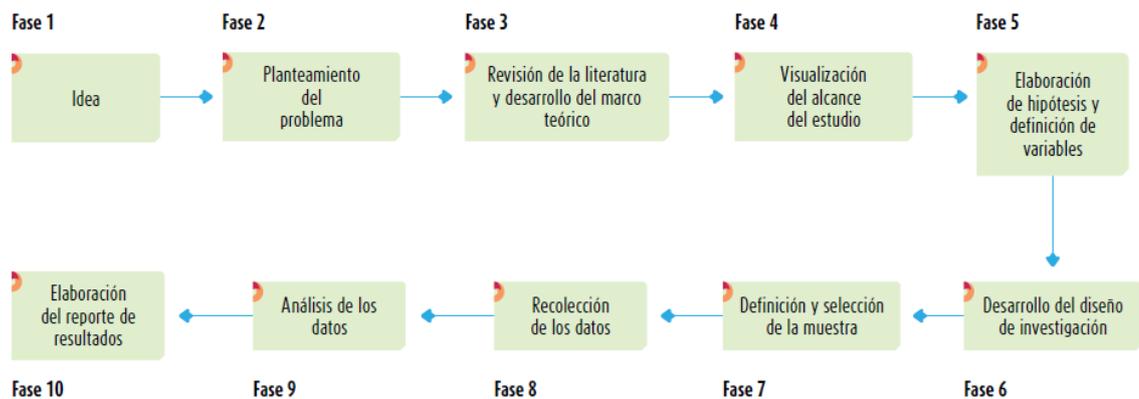


Figura 3.1: Fases del proceso cuantitativo. Tomado de "Metodología de la investigación", por Hernández Sampieri et al., 2014, México, D.F.: McGraw-Hill Education

Por otro lado, el método de investigación es deductivo, se partió del hecho de que la búsqueda y provisión de productos para la construcción es un proceso que está limitado a las fuentes de los sujetos involucrados (profesionales y proveedores en el ámbito de la construcción), es decir, al conocimiento que cada uno tiene sobre dónde encontrar el producto requerido o cómo ofertarlo, dependiendo del caso. Por lo tanto, mucho de la oferta y la demanda de productos para la construcción queda insatisfecha sin conocimiento de los sujetos involucrados.

Ante lo anteriormente mencionado, se llegó a la solución de software propuesta en el presente trabajo de titulación, en el que un sitio en la web que amplía los contactos de los sujetos involucrados basado en sus requerimientos o necesidades de ofertas, es la respuesta para mejorar la oferta y la demanda de productos para la construcción.

Para la elaboración de la aplicación web propuesta, se optó por una metodología ágil de desarrollo de software. Las metodologías ágiles son seleccionadas cuando las variables del proyecto en cuanto a tiempo y recursos disponibles son limitados, el cual fue el caso de este proyecto. Lo mencionado anteriormente, se explica en base a lo que Cabot Sagraera (2013) menciona: “Para determinados proyectos, la rigidez y la meticulosidad impuesta por los métodos de desarrollo actuales puede ser excesiva; es posible que no sea realmente necesario hacer todas las actividades propuestas y/o con el nivel de detalle requerido por el método” (p. 194).

Para este proyecto de titulación se eligió seguir el marco de referencia *Scrum*, que es un subconjunto dentro de las metodologías ágiles de desarrollo de software (también conocidas como metodologías *Agile*), y que, por lo tanto, utiliza sus mismos principios al momento de llevar a cabo un proyecto, los cuales Dimes & Jimenez (2015) resume:

En primer lugar, se crea una pila del producto, que es una lista priorizada de las características o funcionalidades que deberá tener el producto, y las cuales se obtienen de los usuarios potenciales (...). Esta pila (...) contiene aquellas características que harán que el producto sea más rentable. Al ser una lista priorizada (...) se debe trabajar en los elementos más importantes o más urgentes de la pila producto. Esto asegurará que no dedique incontables horas o días en aquellas características menos indispensables dejando de lado las más importantes (p. 2).

Con lo dicho anteriormente, se debe aclarar que no se implementó *Scrum* como tal en el ciclo de vida de desarrollo del aplicativo web propuesto, sino que se tomó sus mismos principios, adaptándolos a las circunstancias del proyecto. Esto es posible dado a que *Scrum* es una forma de implementación de las guías y directrices que propone la metodología *Agile*, por lo tanto, puede adaptarse a las necesidades y circunstancias de cada proyecto.

En *Scrum*, normalmente se planifica y divide las tareas en equipos de trabajo, sin embargo, en la adaptación de *Scrum* para este proyecto, se trabajó

con las limitaciones de un único miembro para la resolución de todas las tareas planteadas en los análisis iniciales y posteriores, siendo la excepción cuando se recibía retroalimentación de los usuarios finales del producto.

Como se explica en párrafos anteriores, se hizo una adaptación de *Scrum* para el ciclo de desarrollo del aplicativo web propuesto. En esta adaptación se siguieron los mismos parámetros que propone el marco referencial, siendo la única diferencia, como ya se explicó anteriormente, la de que solo existe un único planificar y desarrollador.

La adaptación de *Scrum* para este proyecto, se hizo siguiendo los siguiente pasos: primero, se analizaron los requerimientos obtenidos en la fase levantamiento de información y con ellos se creó la pila del producto, en orden de priorización; luego, se determinaron los ciclos iterativos que tendría el desarrollo de la aplicación en base a la pila del producto y la priorización de sus elementos; finalmente, se determinaron los tiempos en los que se evaluaría el progreso de la aplicación y el cómo se organizaría el comienzo y el fin de cada iteración.

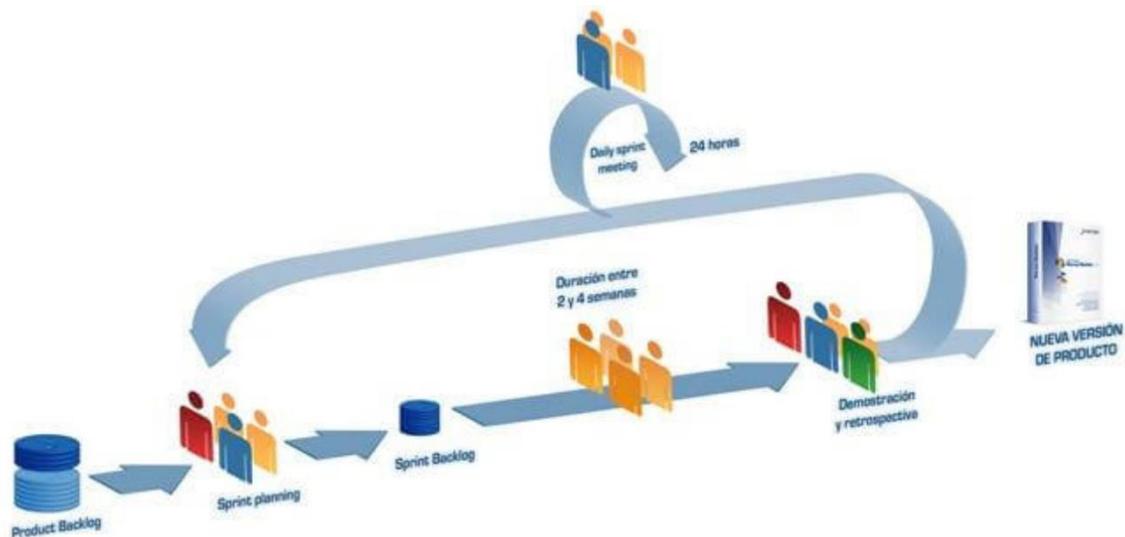


Figura 3.2: Proceso y roles de Scrum.

3.1.2 Población, muestra y muestreo

La población a la que se enfoca la investigación, y a la que se dirige en única instancia el aplicativo web propuesto, es la de proveedores de materiales y profesionales de la construcción en Guayaquil. Es así que, tomando como referencia los datos de la Cámara de la Construcción de Guayaquil, en donde se agrupan un gran número de profesionales dedicados a la construcción, como arquitectos, ingenieros civiles, eléctricos, etc., así como también a los más importantes proveedores de materiales para la construcción, se obtiene que existe un número estimado de 35000 personas, entre profesionales y proveedores.

Basado en los números anteriormente mencionados, para la elaboración de esta investigación, y como parte del levantamiento de información para la definición de los requerimientos funcionales y no funcionales, se estableció trabajar una muestra de 35 profesionales de la construcción y 15 proveedores, suficientes para obtener una opinión general.

Conviene señalar que el muestreo se lo realizó de manera aleatoria, basándose únicamente en que, en el caso de los profesionales de la construcción, que estos tengan entre sus tareas habituales la búsqueda y pedido de materiales para sus obras, y que, en el caso de los proveedores, estos se dediquen exclusivamente a la venta y distribución de materiales para la construcción.

3.1.3 Técnicas e instrumento para la recolección de información

Como técnica para la recolección de datos, se optó por realizar una encuesta a la muestra seleccionada en cada población estudiada en esta investigación. Se trabajó con dos formatos, uno para profesionales y otro para proveedores que trabajen con productos para la construcción. Las preguntas realizadas y las opciones de respuesta fueron establecidas en base a los objetivos de la investigación y el alcance del mismo.

Como instrumento, se eligió utilizar la herramienta web 'Formularios', que es desarrollada y provista por Alphabet Inc., a través de su principal

subsidiaria: Google. Esta herramienta, permite crear formularios personalizados y dinámicos en la web (como informes de asistencia, invitaciones, encuestas, etc.) y a la vez permite recopilar y organizar todo tipo de información en tiempo real.

La encuesta realizada a profesionales de la construcción consta de siete preguntas y la de proveedores seis (ver Anexo 1 y 2). En cada encuesta se evalúa la situación actual del intercambio de información de productos para la construcción y la aceptación de la presente solución web. Además, como método de validación, se pidió el correo electrónico de cada participante.

3.1.4 Herramienta para procesamiento de información

Para el procesamiento de los datos obtenidos en la encuesta, se eligió utilizar las herramientas que provee por el mismo software en línea 'Formularios', como se lo menciona en el punto 3.1.3, el instrumento posee un procesador automático de los datos que son recolectados durante la encuesta.

Para el análisis de los datos (punto 3.2) se utilizó los mismos gráficos estadísticos autogenerados por el instrumento de recolección de datos, que constan de barras o anillos que representan los datos recolectados.

3.2 Análisis de resultados

A continuación, se hará el análisis de los datos obtenidos en la encuesta realizada a profesionales y proveedores en el ámbito de la construcción.

3.2.1 Encuesta a profesiones de la construcción

Pregunta 1

¿Por qué medios usted se entera de los lugares donde los productos (materiales de construcción) son más baratos o están en oferta?

35 respuestas

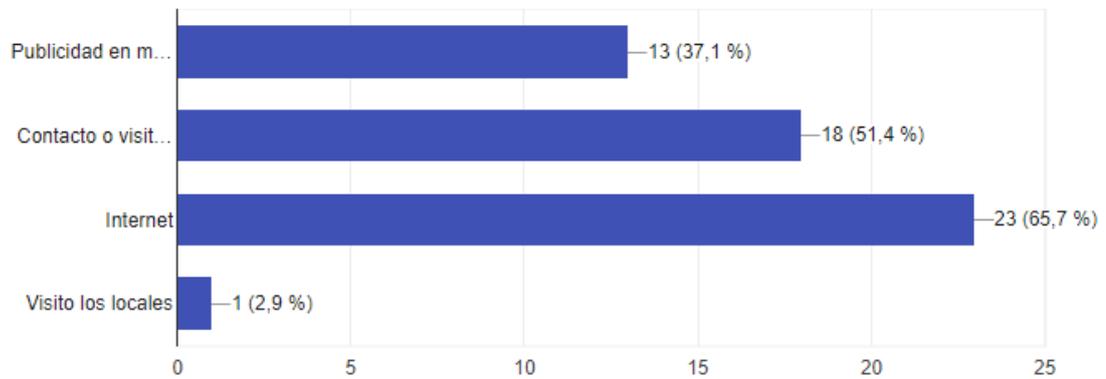


Figura 3.3: Pregunta 1 y gráfico. Encuesta a profesionales: medios de información

Los resultados de esta pregunta muestran un comportamiento asociado a la realidad actual de las comunicaciones, en donde el internet predomina como medio informativo. Se puede observar también, el poco interés en la visita de locales en donde se encuentran los productos. Dado que el internet presenta una opción más cómoda para el usuario, el aplicativo web se presenta como una alternativa bastante aceptable.

Pregunta 2

Si utiliza/utilizara internet ¿cuál de estos espacios utiliza/utilizaría más frecuentemente para enterarse de promociones y precios de productos (materiales de construcción)?

33 respuestas

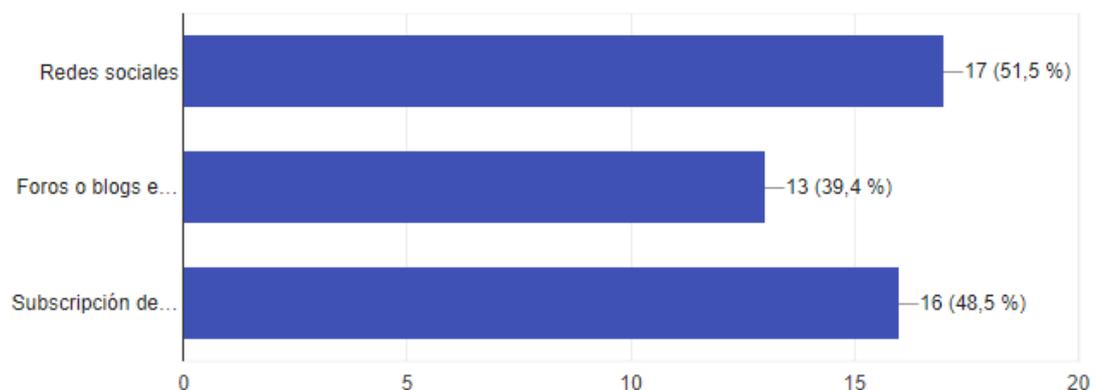


Figura 3.4: Pregunta 2 y gráfico. Encuesta a profesionales: utilización de internet.

La interacción en redes sociales predomina en internet, son el medio más utilizado para comunicar directamente una idea o promocionar un servicio o producto. La aplicación web propuesta se basa en las ideas de interacción en redes sociales.

Pregunta 3

Para la elaboración de presupuestos, los costos de los productos (materiales de la construcción) los obtiene mediante:

35 respuestas

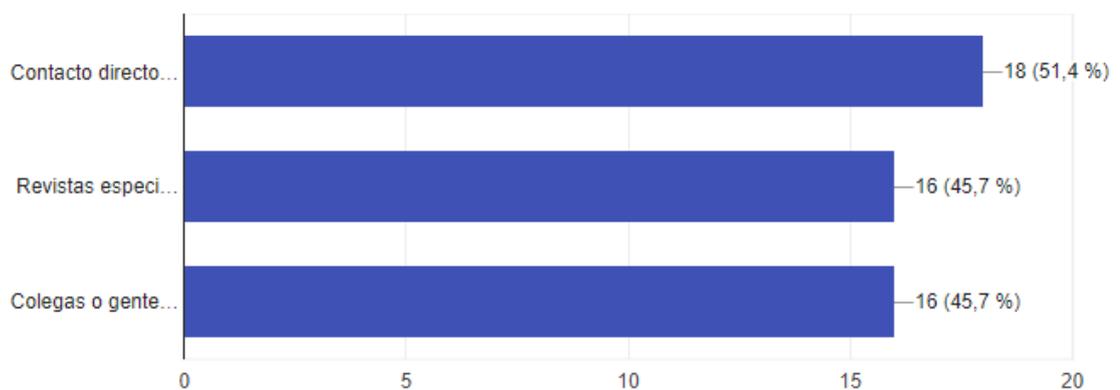


Figura 3.5: Pregunta 3 y gráfico. Encuesta a profesionales: costos de productos

De los resultados a esta pregunta se obtiene que, el comportamiento más habitual al momento de buscar información sobre el costo de un producto, es el de contacto directo con los proveedores. Esto concuerda con la idea de la aplicación, la cual es intentar mejorar la comunicación entre profesionales y proveedores.

Pregunta 4

Defina el nivel de dificultad que usted tiene para conocer los precios de cada uno de los productos (materiales de construcción) que tiene que adquirir para iniciar un trabajo de construcción.

35 respuestas

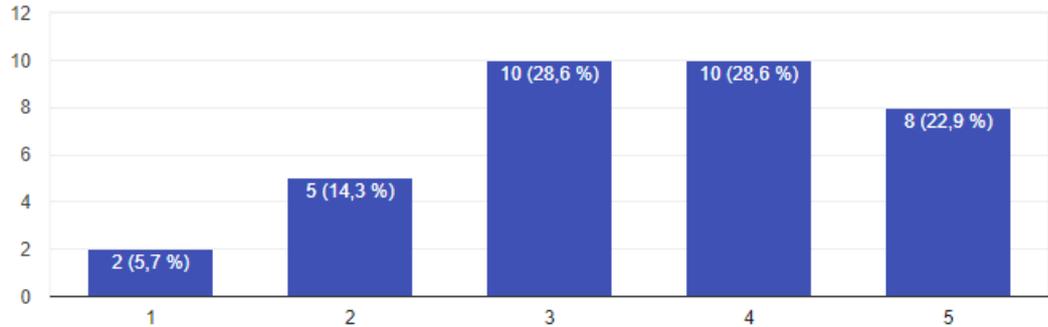


Figura 3.6: Pregunta 4 y gráfico. Encuesta a profesionales: nivel de dificultad.

El nivel de dificultad puede variar dependiendo de la cantidad de contactos con proveedores que un profesional pueda tener. Según las respuestas obtenidas en esta pregunta, el nivel de dificultad en la mayoría de los casos va de un punto intermedio a ser bastante difícil. Esto representa para la aplicación, el problema a resolver, que lo hace estableciendo un lugar único en donde encontrar la información requerida.

Pregunta 5

¿Considera usted que sus fuentes son lo suficientemente eficientes para hallar los productos que usted busca?

35 respuestas



Figura 3.7: Pregunta 5 y gráfico. Encuesta a profesionales: fuentes de información.

Existe una necesidad de fuentes de información para los profesionales, lo que se ve reflejado en los resultados de esta pregunta. La aplicación web propuesta pretende solventar esta necesidad.

Pregunta 6

¿Cree usted que los avances tecnológicos le han ayudado a realizar más eficientemente la búsqueda de los productos que requiere?

35 respuestas

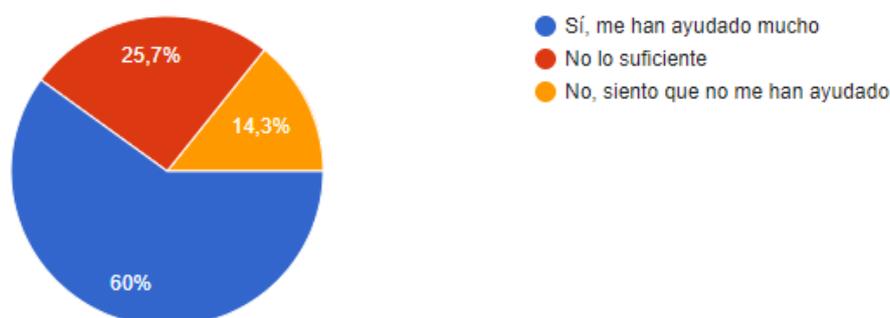


Figura 3.8: Pregunta 6 y gráfico. Encuesta a profesionales: fuentes de información.

Los avances tecnológicos mejoran de manera significativa los procesos que realizan los profesionales. Sin embargo, no en todas las áreas los avances evolucionan de la misma manera. Si bien los resultados de la encuesta muestran un resultado positivo en cuanto a la ayuda que los avances tecnológicos han brindado a los profesionales, también podemos notar que estos no satisfacen a todos, por lo que la mejora en los procesos de búsqueda de información sobre productos, sigue siendo una razón aceptable para el desarrollo de una aplicación web como la que se propone en este trabajo de titulación.

Pregunta 7

¿Utilizaría un sitio web en donde pudiera realizar peticiones de los productos que requiere directamente con los proveedores?

35 respuestas



Figura 3.9: Pregunta 7 y gráfico. Encuesta a profesionales: aplicación web.

Un sitio web en donde se pueda realizar requerimientos de productos para la construcción es muy aceptado por la comunidad de profesionales de esa área.

3.2.2 Encuesta a proveedores de productos para la construcción

Pregunta 1

¿A través de qué medios usted oferta sus productos al consumidor?

15 respuestas

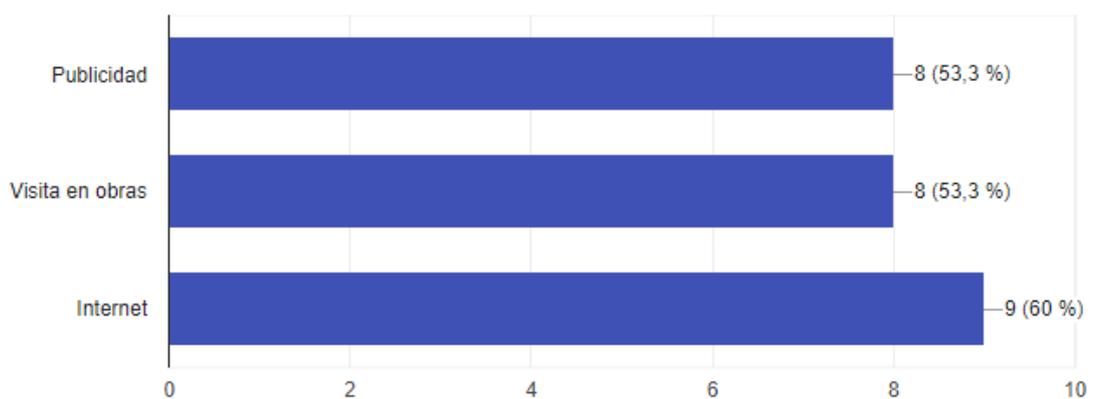


Figura 3.10: Pregunta 1 y gráfico. Encuesta a proveedores: medios de comunicación.

Las repuestas de esta pregunta muestra una tendencia, cada vez más creciente, de la oferta de productos a través de internet. Sin embargo, sigue

estando presente la publicidad y las visitas en obras como un constante medio para ofertar productos.

Pregunta 2

En caso de utilizar internet ¿Cuál de estos espacios utiliza/utilizaría más frecuentemente para promocionar sus productos?

15 respuestas

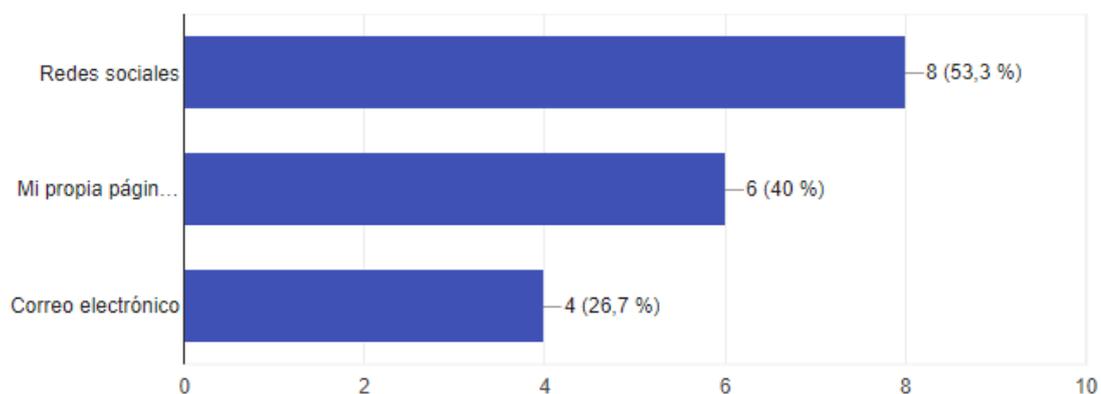


Figura 3.11: Pregunta 2 y gráfico. Encuesta a proveedores: internet.

Como en el caso de los profesionales, en internet, el medio de comunicación masivo, que la mayoría de proveedores considera más eficiente, son las redes sociales. Seguido, están los sitios web propios que cada proveedor puede tener, sin embargo, en la mayoría de los casos, son utilizados en conjunto con las redes sociales.

Pregunta 3

¿Qué nivel de dificultad usted tiene para hallar a sus clientes potenciales?

15 respuestas

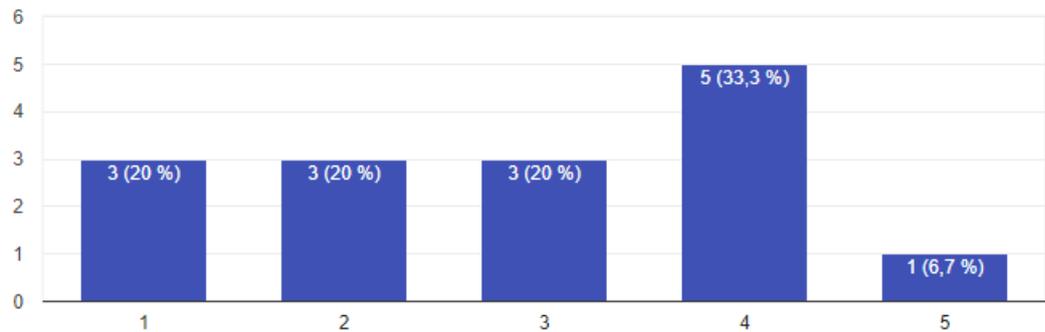


Figura 3.12: Pregunta 3 y gráfico. Encuesta a proveedores: clientes potenciales.

¿Por qué?

15 respuestas

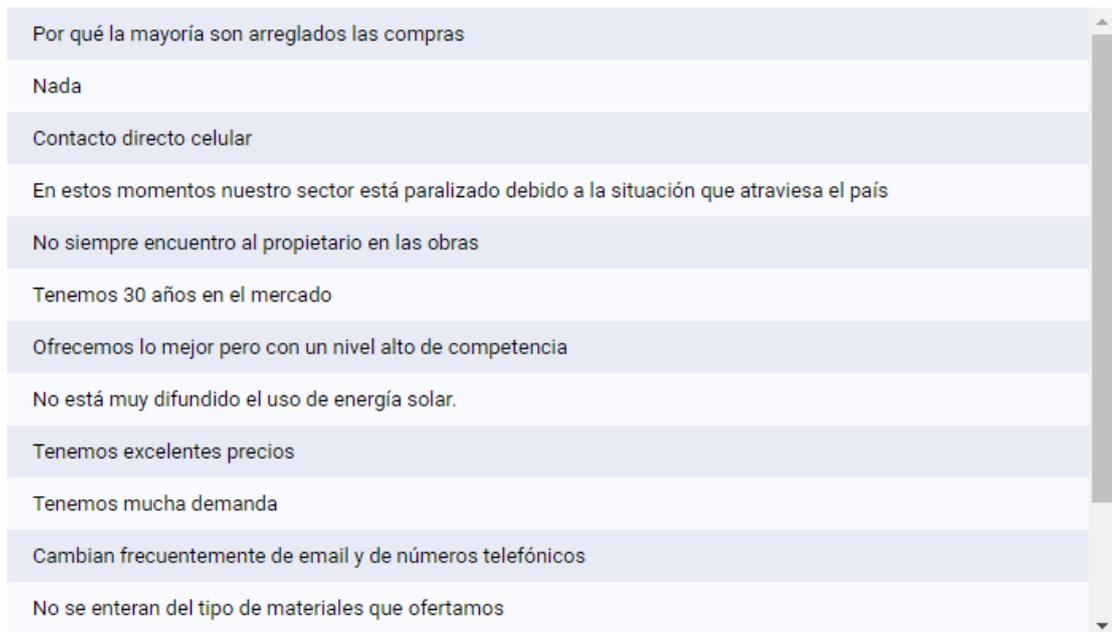


Figura 3.13: Razones de la pregunta 3.

La dificultad que tienen los proveedores al momento de hallar a sus clientes potenciales varía dependiendo de las circunstancias de cada proveedor, es decir, un proveedor conocido tiene más oportunidades que uno que no lo sea. Sin embargo, se puede decir que es un proceso que demanda muchos recursos y que en la mayoría de los casos (basándose en las respuestas a esta pregunta) no es tan fácil.

Pregunta 4

¿Cómo entrega a sus clientes la cotización de sus productos?

15 respuestas

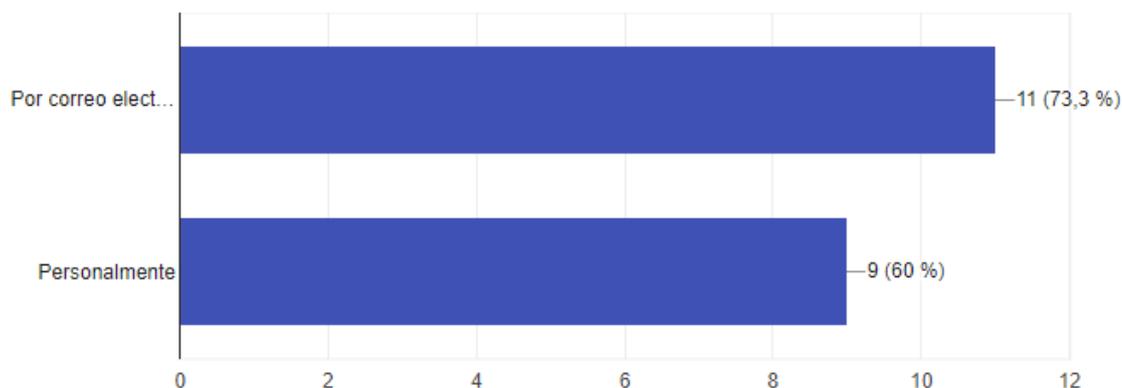


Figura 3.14: Pregunta 4 y gráfico. Encuesta a proveedores: cotización de productos.

La tendencia de realizar la mayor parte de los procesos a través de internet (en este caso, correo electrónico), es lo que se puede observar en los resultados de esta pregunta. También, se observa que la mayoría utiliza ambos medios, lo que supone un comportamiento de seguir realizando el proceso de cotización personalmente.

Pregunta 5

¿Utilizaría un sitio web que le brindara la oportunidad de ofertar directamente con sus clientes potenciales a través de la realización de cotizaciones virtuales?

15 respuestas

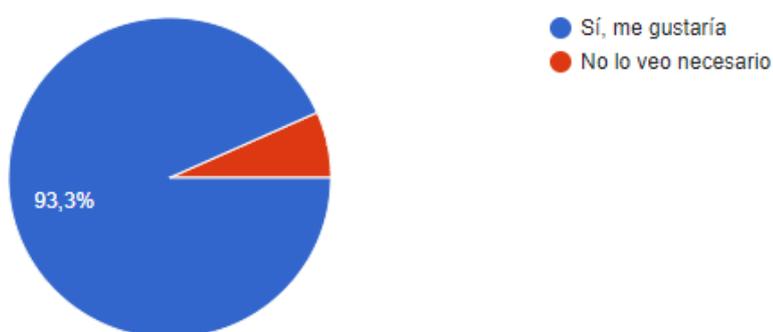


Figura 3.15: Pregunta 5 y gráfico. Encuesta a proveedores: aceptación de aplicativo.

Se observa una amplia aceptación de un sitio web en donde se pudiera realizar lo que la pregunta describe, lo que justifica este proyecto.

Pregunta 6

En ese sitio web ¿Cómo le gustaría presentar sus cotizaciones a sus potenciales clientes?

15 respuestas

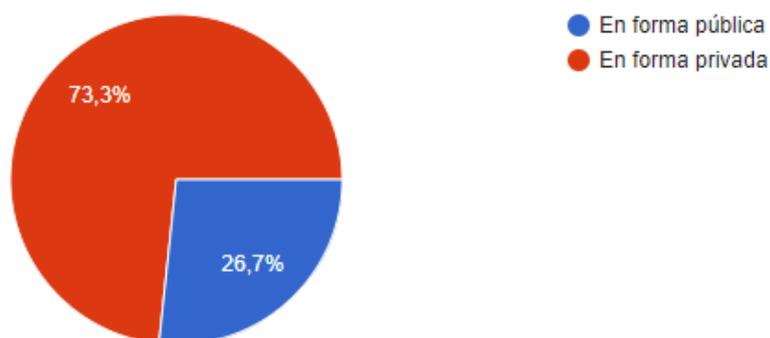


Figura 3.16: Pregunta 6 y gráfico. Encuesta a proveedores: requerimiento en aplicación.

La forma en la que la mayoría de proveedores desea realizar una cotización, es de forma privada. Esto supone un requerimiento a tomar en cuenta para la aplicación.

CAPÍTULO IV: PROPUESTA TECNOLÓGICA

En este capítulo se detalla en profundidad la propuesta tecnológica y como esta fue construida. Se describe el entorno de desarrollo utilizado para la codificación del software y se revisa la estructuración y organización lógica de la aplicación. Además, se describe la funcionalidad de cada módulo de la aplicación, mostrando sus respectivas pantallas.

4.1 Herramientas de desarrollo

Para el desarrollo de la aplicación web propuesta, se escogió utilizar herramientas de software que funcionen bajo el sistema operativo Windows 10, con el motivo de trabajar en un solo ambiente para la codificación y las pruebas.

En los puntos a continuación, se describirán las herramientas de software escogidas y el motivo de su selección.

4.1.1 Visual Studio Code

Editor de código desarrollado y provisto por Microsoft para sistemas operativos Windows, iOS y distribuciones Linux. Es de código abierto y por tanto personalizable. Incluye soporte para la depuración y refactorización de código, además de complementos para la revisión de sintaxis y autocompletado de código. Dispone de herramientas de control para el versionado a través de Git.

Se seleccionó esta herramienta porque facilita el manejo del versionado del software a través de Git, sin tener que recurrir a la apertura de una consola y el manejo de comandos.

4.1.2 Bash de Ubuntu en Windows 10

Adaptación del sistema operativo Linux Ubuntu y su consola de comandos para sistemas operativos Windows en su versión 10. Permite ejecutar comandos típicos de Unix en Windows.

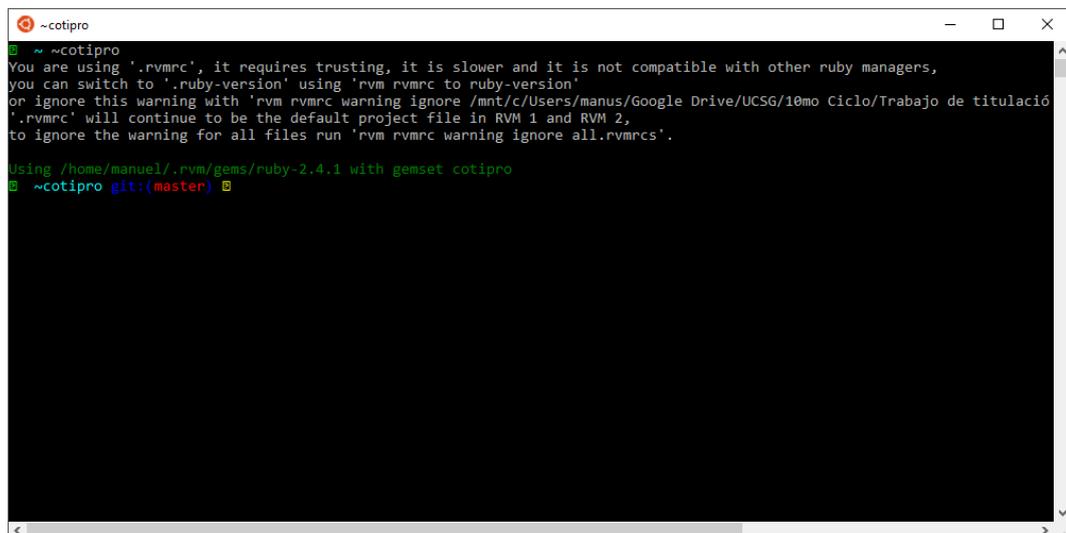
El motivo para utilizar esta herramienta es que permite instalar aplicaciones de Linux Ubuntu a través de su manejador de paquetes por defecto apt-get, lo que brinda la oportunidad de utilizar herramientas de desarrollo exclusivas de este sistema operativo en Windows 10.

4.1.3 MongoDB Community Edition

Motor de base de datos no relacional (NoSQL) orientado a documentos. Provee de un conjunto de comandos que se instalan en el sistema operativo, para el control y manejo de colecciones de tipo JSON.

Se utilizó esta herramienta en conjunto con Mongoid, un software de mapeo objeto-relacional (ORM por sus siglas en inglés), diseñado para ambientes en donde se utilice el lenguaje de programación Ruby.

Se decidió utilizar esta versión de MongoDB porque provee de las herramientas necesarias para satisfacer los requerimientos de la aplicación.



```
~ -cotipro
└─$ ~cotipro
You are using '.rvmrc', it requires trusting, it is slower and it is not compatible with other ruby managers,
you can switch to '.ruby-version' using 'rvm rvmrc to ruby-version'
or ignore this warning with 'rvm rvmrc warning ignore /mnt/c/Users/manus/Google Drive/UCSG/10mo Ciclo/Trabajo de titulació
'.rvmrc' will continue to be the default project file in RVM 1 and RVM 2,
to ignore the warning for all files run 'rvm rvmrc warning ignore all.rvmrcs'.

Using /home/manuel/.rvm/gems/ruby-2.4.1 with gemset cotipro
└─$ ~cotipro git:(master) ─
```

Figura 4.1: Consola de comando Bash de Ubuntu en Windows 10

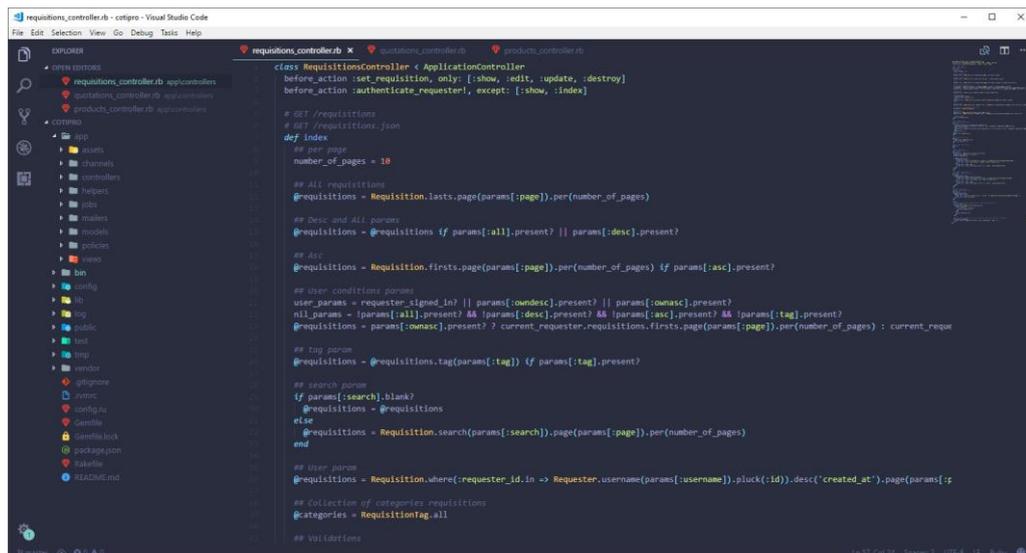


Figura 4.2: Entorno de desarrollo Visual Studio Code

4.2 Lenguaje de programación y *framework*

El código fuente de la aplicación web propuesta fue escrito en el lenguaje de programación orientado a objetos Ruby. La sintaxis de Ruby está inspirada en Python y Perl (otros lenguajes de programación), y su filosofía, según Yukihiro Matsumoto (creador del lenguaje), es la productividad y ‘diversión’ del programador. Se eligió este lenguaje porque el *framework* con el que se trabajó el desarrollo de la aplicación, *Ruby on Rails* (RoR), utiliza, como su nombre da a suponer, Ruby.

Ruby sigue el principio de la menor sorpresa, es decir, si alguien es un usuario con experiencia en otro lenguaje orientados objetos y decide intentar programar en Ruby, encontrará en éste pocas diferencias a la hora de codificar, por lo que curva de aprendizaje es bastante baja. Esta adaptabilidad para el programador fue otro motivo para su selección en este proyecto.

En cuanto al *framework*, Ruby on Rails, también conocido como RoR o simplemente Rails, permite crear aplicaciones web siguiendo el paradigma o patrón Modelo Vista Controlador (MVC). Fue elegido para el desarrollo de la aplicación por sus principios de simplicidad y mínima configuración, además de utilizar la ‘metaprogramación’ característica del lenguaje Ruby, lo que permite crear los módulos base de una aplicación a través de la ejecución de comandos auto-generadores.

Los principios que definen al *framework* y que lo hacen uno de los más utilizados (ver figura 4.3), es el de ‘No te repitas’ (DRY por sus siglas en inglés) y el de ‘Convención sobre Configuración’. El principio de ‘No te repitas’ se refiere a que no es necesario definir un bloque de código o función más de una vez, esto es porque el *framework* es de pila completa, es decir, sus componentes están complemente integrados y no necesitan un puente entre ellos. Con ‘Convención sobre Configuración’ se refiere a que la configuración solo necesaria cuando esta no es convencional; esto permite, por ejemplo, crear modelos de una aplicación de manera más sencilla, es decir, si se crea el modelo ‘Producto’, el *framework* por convención sabe que ese modelo pertenece a la colección (o tabla si se tratara de una base de datos relacional) ‘productos’, si el nombre de este último es diferente, entonces es necesaria una configuración.

En términos legales de uso, Rails es una gema (librería de Ruby) que puede ser adquirida (descargada) de manera gratuita a través del distribuidor de paquetes oficiales de Ruby: RubyGems. Es de código abierto y su utilización está bajo la licencia MIT.

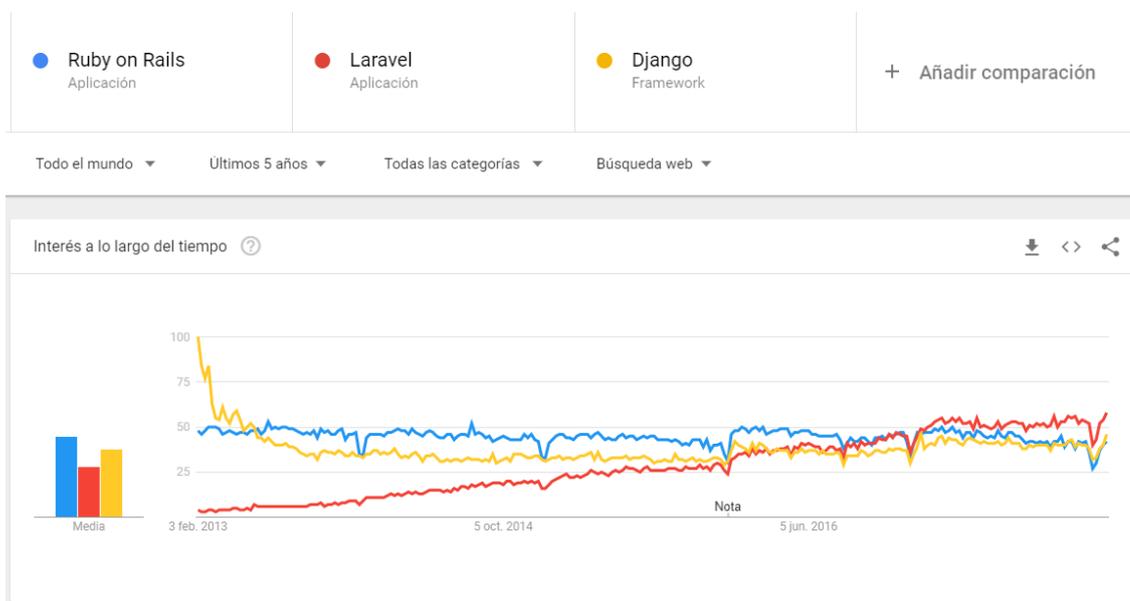


Figura 4.3: Utilización del framework Ruby on Rails en comparación con sus similares en los últimos cinco años.

4.3 Estructura y organización de la aplicación

La estructura de la aplicación web propuesta está basada en la misma que se autogenera al crear una nueva aplicación utilizando el *framework* Ruby on Rails, a través del comando `rails new <nombre de la aplicación>` (ver figura 4.4). La estructura de Rails se compone de nueve carpetas y siete archivos en su forma inicial (autogenerada), siendo la carpeta que lleva como nombre *app* la principal, y en donde se encuentra en mayor parte la lógica y organización de la aplicación.

En la carpeta *app* existe a su vez ocho subcarpetas (ver figura 4.5) que, como se dijo antes, concentran la mayor parte de la lógica de la aplicación, debido a que aquí se encuentran los modelos, controladores y las vistas que son ejecutadas en el servidor (tanto en desarrollo como en producción) en sus respectivas carpetas.

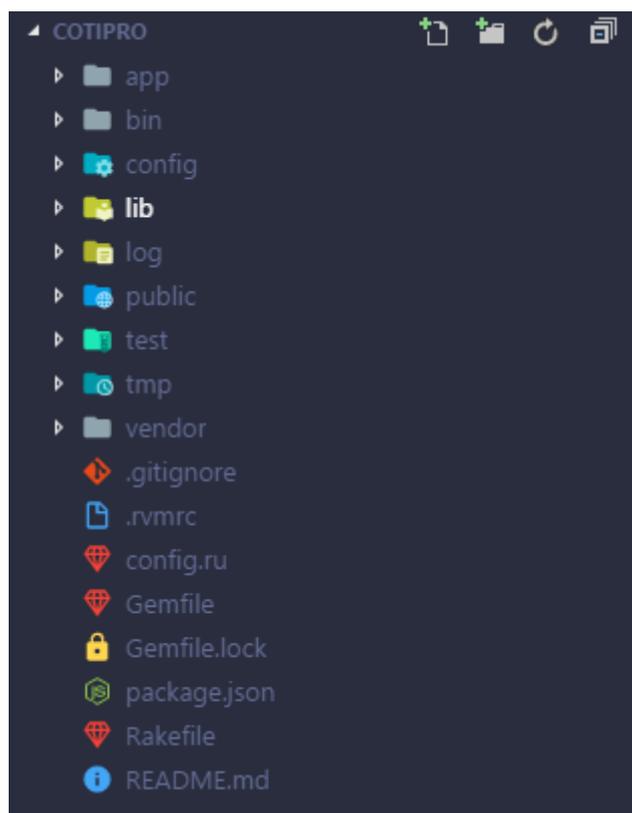


Figura 4.4: Estructura de la aplicación *cotipro* autogenerada por Ruby on Rails

En los puntos a continuación, se describirá a detalle la función y contenido de las carpetas más significantes de esta estructura, las necesarias para comprender como funciona la aplicación web propuesta.

4.3.1 La carpeta *app*

En esta carpeta se concentra la lógica y organización de la aplicación. Los modelos, controladores y vistas que son parte del paradigma MVC, se organizan en las carpetas *models*, *controllers* y *views* respectivamente.

Los modelos que son contenidos dentro de la carpeta *models*, son, para el caso específico de esta aplicación, y porque se utiliza el ORM Mongoid, la representación de las colecciones de la base de datos no relacional MongoDB. Cada modelo dentro de esta carpeta es un archivo de tipo Ruby en donde son definidas sus validaciones y métodos respectivos, además de la definición de los campos de la colección, siendo esto último algo particular del ORM utilizado.

Los controladores en la carpeta *controllers* son archivos de tipo Ruby en donde se definen las variables que almacenan las consultas a la base de datos según los parámetros enviados desde la vista. Cada controlador contiene las definiciones respectivas para cada vista concebida en las rutas definidas de la aplicación (ver punto 4.3.2), así como también validaciones de sesiones de usuario.

La carpeta *views* contiene archivos HTML que están dentro de otras carpetas con el nombre correspondiente a los controladores existentes de la aplicación, es decir, por ejemplo: el controlador '*product*' tendría sus vistas en la ruta '*views/products*'. Cada archivo HTML viene formateado con la extensión '*.erb*', lo que permite escribir en este tipo de archivos código en lenguaje Ruby.

Otras carpetas importantes son *assets*, *helpers*, y *mailers*. La carpeta *assets* contiene las imágenes (estáticas), estilos CSS, y el JavaScript de la aplicación; la carpeta *helpers* contiene métodos de ayuda para cada controlador escritos en lenguaje Ruby, su principal uso es para resumir código y volverlo reutilizable; la carpeta *mailers* contiene los métodos que permiten configurar los datos que son enviados por correo electrónico a través de la aplicación.

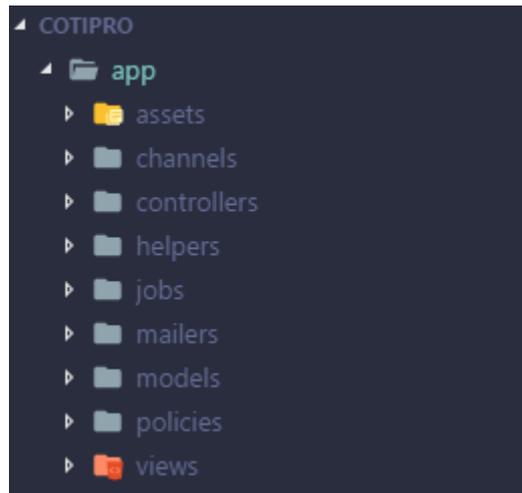


Figura 4.5: Carpeta *app* dentro de la estructura de *Cotipro*

4.3.2 La carpeta *config*

Esta carpeta, como su nombre sugiere, contiene los archivos de configuración de la aplicación, siendo los más relevantes aquellos en donde se definen las rutas (URLs), la forma de conexión a la base datos, las variables de entorno para desarrollo y producción, los inicializadores de librerías y el idioma.

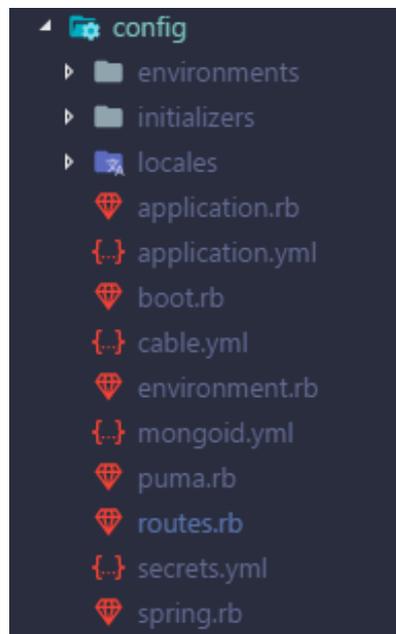


Figura 4.6: Carpeta *config* dentro de la estructura de *Cotipro*

En el desarrollo de la aplicación, la configuración que sufre más cambios durante los ciclos iterativos es la de las rutas, que se encuentra definida dentro de *routes.rb*. En este archivo de configuración se utilizan

palabras reservadas de Rails para definir las rutas, basadas en la arquitectura REST. Tal y como se puede observar en las figuras 4.7 y 4.8, la palabra reservada *resources* genera todas rutas que permiten realizar la creación, lectura, edición y eliminación de un registro de la base de datos a través del controlador correspondiente.

```

routes.rb x
1  Rails.application.routes.draw do
2    root 'home#index'
3
4    resources :requisitions do
5      resources :quotations, only: [:show, :create]
6    end
7    resources :quotations, only: [:edit, :update, :destroy]
8
9    resources :quotations, except: [:show] do
10     resources :comments, only: [:create, :update, :destroy]
11   end
12
13   resources :products
14

```

Figura 4.7: Archivo de configuración de rutas routes.rb

```

~cotipro git:(master) rails routes
Prefix Verb  URI Pattern                               Controller#Action
root GET    /                                           home#index
requisition_quotations POST /requisitions/:requisition_id/quotations(.:format) quotations#create
requisition_quotation GET /requisitions/:requisition_id/quotations/:id(.:format) quotations#show
requisitions GET /requisitions(.:format) requisitions#index
requisitions POST /requisitions(.:format) requisitions#create
new_requisition GET /requisitions/new(.:format) requisitions#new
edit_requisition GET /requisitions/:id/edit(.:format) requisitions#edit
requisition GET /requisitions/:id(.:format) requisitions#show
requisition PATCH /requisitions/:id(.:format) requisitions#update
requisition PUT /requisitions/:id(.:format) requisitions#update
requisition DELETE /requisitions/:id(.:format) requisitions#destroy
edit_quotation GET /quotations/:id/edit(.:format) quotations#edit
quotation PATCH /quotations/:id(.:format) quotations#update
quotation PUT /quotations/:id(.:format) quotations#update
quotation DELETE /quotations/:id(.:format) quotations#destroy
quotation_comments POST /quotations/:quotation_id/comments(.:format) comments#create
quotation_comment PATCH /quotations/:quotation_id/comments/:id(.:format) comments#update

```

Figura 4.8: Ejecución del comando rails routes que permite ver las rutas configuradas en routes.rb. En la figura no se muestran todas las rutas.

Las configuraciones de la conexión a la base de datos, variables de entorno y traducciones del inglés al español, son definidas en archivos de tipo YAML (acrónimo recursivo que significa *YAML Ain't Markup Language*), en donde se establece la representación de los datos que sirven para que la aplicación pueda ejecutar ciertas tareas o interpretar ciertos datos. En la figura

4.9, se puede observar, como ejemplo de configuración en YAML, el archivo *mongoid.yml*, que establece la conexión a la base de datos.

```
mongoid.yml
1  development:
2    # Configure available database clients. (required)
3    clients:
4      # Defines the default client. (required)
5      default:
6        # Defines the name of the default database that Mongoid can connect to.
7        # (required).
8        database: cotipro_development
9        # Provides the hosts the default client can connect to. Must be an array
10       # of host:port pairs. (required)
11       hosts:
12         - localhost:27017
13       options: ...
103
104 # Configure Mongoid specific options. (optional)
105 options: ...
147 test:
148   clients:
149     default:
150       database: cotipro_test
151       hosts:
152         - localhost:27017
153       options:
154         read:
155           mode: :primary
156           max_pool_size: 1
157 production:
158   clients:
159     default:
160       uri: <%= ENV['MONGODB_URI'] %>
161
162   options:
163     connect_timeout: 15
```

Figura 4.9: Resumen del archivo *mongoid.yml*, que establece las conexión a la base de datos en los ambiente de desarrollo, prueba y producción.

4.3.3 El archivo *Gemfile*

En el archivo *Gemfile* se definen las componentes de software o librerías que la aplicación va a utilizar en sus ambientes de desarrollo, pruebas y producción. En el ecosistema de Ruby las librerías son conocidas como *gems* (gemas en español) y de ahí el motivo del nombre de este archivo.

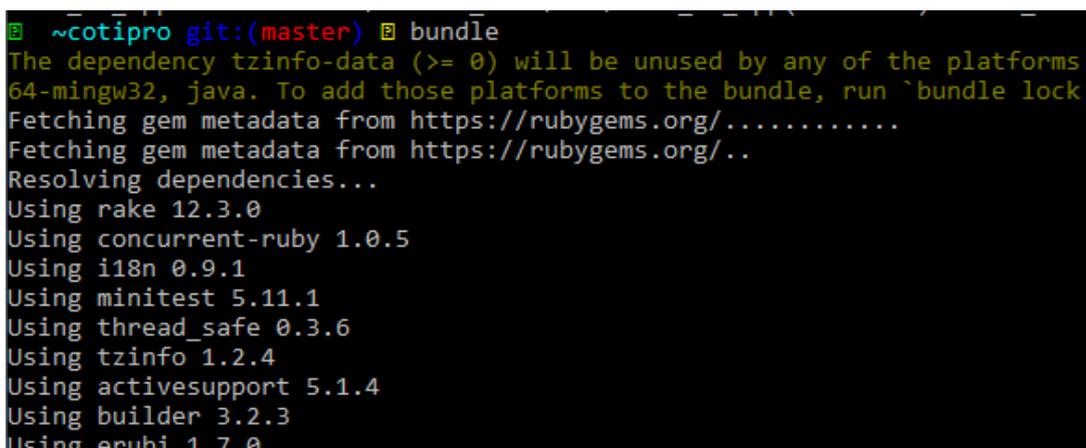
Como se puede apreciar en la figura 4.10, las gemas son definidas con una simple llamada por su nombre. Esto se debe a que, con la ayuda de otra gema, llamada *bundler*, al ser ejecutada desde la consola de comandos (ver figura 4.11), leerá el archivo *Gemfile* y comenzará a descargar y agregar al proyecto las gemas declaradas. La gema *bundler* consigue las gemas directo desde el repositorio oficial de gemas de Ruby, RubyGems.org.

Por defecto, al crear un nuevo proyecto con Ruby on Rails, el archivo *Gemfile* vendrá con un conjunto de gemas declaradas, la del mismo *framework* entre ellas. Sin embargo, para el desarrollo de esta aplicación, se tuvieron que agregar gemas que permitieran alcanzar los objetivos de desarrollar los requerimientos funcionalidades deseados. En la figura 4.12 se puede observar cuales gemas fueron agregadas.



```
1 source 'https://rubygems.org'
2
3 git_source(:github) do |repo_name|
4   repo_name = "#{repo_name}/#{repo_name}" unless repo_name.include?("/")
5   "https://github.com/#{repo_name}.git"
6 end
7
8 ruby '2.4.1'
9
10 gem 'mongoid', '~> 6.1.0'
11 # Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
12 gem 'rails', '~> 5.1.4'
13 # Use Puma as the app server
14 gem 'puma', '~> 3.7'
15 # Use SCSS for stylesheets
16 gem 'sass-rails', '~> 5.0'
17 # Use Uglifier as compressor for JavaScript assets
18 gem 'uglifier', '>= 1.3.0'
19 # See https://github.com/rails/execjs#readme for more supported runtimes
20 # gem 'therubyracer', platforms: :ruby
21
```

Figura 4.10: Primeras líneas del archivo *Gemfile*. A partir de la línea 10 la definición de las gemas.



```
~cotipro git:(master) bundle
The dependency tzinfo-data (>= 0) will be unused by any of the platforms
64-mingw32, java. To add those platforms to the bundle, run `bundle lock
Fetching gem metadata from https://rubygems.org/.....
Fetching gem metadata from https://rubygems.org/..
Resolving dependencies...
Using rake 12.3.0
Using concurrent-ruby 1.0.5
Using i18n 0.9.1
Using minitest 5.11.1
Using thread_safe 0.3.6
Using tzinfo 1.2.4
Using active_support 5.1.4
Using builder 3.2.3
Using erubi 1.7.0
```

Figura 4.11: Ejecución del comando *bundle*, que descarga e instala las gemas declaradas en el *Gemfile* desde *rubygems.org*

```

38 # Gemas
39 gem 'mongoid-paranoia'
40 gem 'mongoid-slug'
41 gem 'materialize-sass'
42 gem 'material_icons'
43 gem 'devise'
44 gem 'select2-rails'
45 gem 'chosen-rails'
46 gem 'pundit'
47 gem 'wysiwyg-rails'
48 gem 'mongoid-paperclip'
49 gem 'kaminari-mongoid'
50 gem 'rails_admin', '~> 1.2'
51 gem 'rails_admin_material'
52 gem 'figaro'
53 gem 'aws-sdk', '~> 2.3'
54 gem 'cocoon'
55 gem 'jquery-ui-rails'

```

Figura 4.12: Gemas agregadas al proyecto, en el archivo Gemfile

4.4 Modelado de la base de datos

El modelado de la base de datos se basa en una estructura no relacional, correspondiente al tipo de motor de base de datos seleccionado, en este caso MongoDB. A diferencia de las bases de datos relacionales, en las no relacionales existen colecciones en vez de tablas. Es así que, se detallará a continuación la estructura y los campos de cada colección creada para el funcionamiento de la aplicación web de este trabajo de titulación.

4.4.1 Colección Usuarios

Definida en el modelo *User* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/user.rb`. Almacena a todos los usuarios que se registren en la aplicación. Su modelo sirve de base para los modelos de las colecciones que almacenan y separan a los usuarios en requirentes y proveedores, además de otro modelo para el administrador.

```

1  class User
2    include Mongoid::Document
3    include Mongoid::Paperclip
4
5    devise :database_authenticatable, :registerable,
6           :recoverable, :rememberable, :trackable, :validatable
7
8    has_many :comments
9
10   ## Database authenticatable
11   field :email, type: String, default: ""
12   field :encrypted_password, type: String, default: ""
13
14   ## Recoverable
15   field :reset_password_token, type: String
16   field :reset_password_sent_at, type: Time
17
18   ## Rememberable
19   field :remember_created_at, type: Time
20
21   ## Trackable
22   field :sign_in_count, type: Integer, default: 0
23   field :current_sign_in_at, type: Time
24   field :last_sign_in_at, type: Time
25   field :current_sign_in_ip, type: String
26   field :last_sign_in_ip, type: String
27
28   ## Confirmable
29   field :confirmation_token, type: String
30   field :confirmed_at, type: Time
31   field :confirmation_sent_at, type: Time
32   field :unconfirmed_email, type: String # Only if using reconfirmable
33
34   has_mongoid_attached_file :avatar,
35     styles: {
36       original: '1920x1680>',
37       small: '128x128#',
38       thumb: '180x180',
39       medium: '250x250',
40       large: '500x500>'
41     },
42     default_url: 'user/default.png'
43
44   validates_attachment_content_type :avatar, :content_type => /\Aimage\/.*\Z/
45 end

```

Figura 4.13: Colección de usuarios definida en el modelo *User*

4.4.2 Colección Requirientes

Definida en el modelo *Requester* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/requester.rb`. Almacena a los usuarios de tipo requirente (profesional de la construcción). Tiene una relación de uno a muchos con las requisiciones. Está conectada al modelo *User*.

4.4.3 Colección Proveedores

Definida en el modelo *Provider* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/provider.rb`. Almacena a los usuarios de tipo

proveedor (proveedor de materiales de construcción). Tiene una relación de uno a mucho con las cotizaciones y de la misma forma con los productos. Está conectada al modelo *User*.

```

1 class Requester < User
2   attr_accessor :login
3
4   has_many :requisitions, dependent: :destroy
5
6   devise :confirmable, :authentication_keys => [:login]
7
8   has_many :requisitions, dependent: :destroy
9
10  ## Info
11  field :name, type: String
12  field :username, type: String
13
14  ## is_requester
15  field :is_requester, type: Boolean, default: true
16  field :is_provider, type: Boolean, default: false
17
18  validates :name, presence: true
19  validates :username, presence: true, uniqueness: { conditions: -> { where(deleted_at: nil) }, case_sensitive: false }
20  validates_format_of :username, with: /^[a-zA-Z0-9_\.]*$/, :multiline => true
21  validate :validate_username
22
23  scope :username, -> (username) { where(username: username) }
24
25  def validate_username
26    if User.where(email: username).exists?
27      errors.add(:username, :invalid)
28    end
29  end
30
31  def self.find_first_by_auth_conditions(warden_conditions)
32    conditions = warden_conditions.dup
33    if login = conditions.delete(:login)
34      self.any_of({ username: /#{::Regexp.escape(login)}$/i }, { email: /#{::Regexp.escape(login)}$/i }).first
35    else
36      super
37    end
38  end
39 end

```

Figura 4.14: Colección Requirientes definida en el modelo Requester

```

1 class Provider < User
2   devise :confirmable, :authentication_keys => [:email]
3
4   has_many :products, dependent: :destroy
5   has_many :quotations
6
7   ## Contact info
8   field :name, type: String
9   field :business_name, type: String
10  field :ruc, type: String
11  field :contact_email, type: String
12  field :contact_number_conventional, type: String
13  field :contact_number_cellphone, type: String
14  field :address, type: String
15  field :description, type: String
16
17  ## is_provider
18  field :is_provider, type: Boolean, default: true
19  field :is_requester, type: Boolean, default: false
20
21  validates :name, :business_name, :ruc, :contact_email,
22           :contact_number_conventional, :contact_number_cellphone,
23           :address, presence: true
24  validates :ruc, length: { is: 13 }
25  validates :description, length: { minimum: 5, maximum: 20 }, allow_blank: true
26 end

```

Figura 4.15: Colección Proveedores definida en el modelo Provider

4.4.4 Colección Requisición

Definida en el modelo *Requisition* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/requisition.rb`. Almacena las requisiciones creadas por los requirentes (profesionales de construcción), definidos en el modelado como *requesters*. Tiene una relación con las cotizaciones, que pueden ser varias. Ver figura 4.16.

4.4.5 Colección Productos

Definida en el modelo *Product* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/product.rb`. Almacena los productos ingresados por los proveedores. Tiene una relación con los proveedores. Ver figura 4.17.

4.4.6 Colección Cotizaciones y colección Elementos

En primer lugar, la colección Cotizaciones, que está definida en el modelo *Quotation* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/quotation.rb`, mientras que la colección Elementos, está definida en el modelo *Item*, y está ubicada en la ruta `/app/models/item.rb`. Ambas colecciones están relacionadas por su inherencia en la creación de las cotizaciones. Cada cotización tiene una relación uno a muchos con los elementos. Los elementos hacen referencia al listado de productos que son agregados durante la realización de una cotización. Ver figura 4.18 y 4.19.

4.4.7 Colección Categorías

Definida en el modelo *Requisition_tag* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/requisition_tag.rb`. Almacena las categorías ingresados por el administrador de la aplicación y que pueden ser seleccionadas por un requirente al momento de crear una requisición. Ver figura 4.20.

4.4.8 Colección Comentarios

Definida en el modelo *Comment* dentro de la estructura de la aplicación, ubicada en la ruta: `/app/models/comment.rb`. Almacena los productos los comentarios realizados por los proveedores o requirentes en una cotización. Tiene una relación con usuarios y las cotizaciones. Ver figura 4.21.

```
1 class Requisition
2   include Mongoid::Document
3   include Mongoid::Paranoia
4   include Mongoid::Slug
5   include Mongoid::Timestamps
6
7   belongs_to :requester
8   has_many :quotations, dependent: :destroy
9
10  field :title,      type: String
11  field :description, type: String
12  field :tag,       type: Array
13  slug :title, max_length: 24
14
15  scope :tag, -> (tag) { where(tag: tag) }
16  scope :lasts, -> { where(nil).desc('created_at') }
17  scope :firsts, -> { where(nil).asc('created_at') }
18
19  def self.search(search)
20    if search
21      any_of({title: /#{search}/i}, {description: /#{search}/i})
22    end
23  end
24
25  validates :title, :description, :tag, presence: true
26  validates :description, length: { minimum: 20 }
27
28  before_validation do |model|
29    model.tag.reject!(&:blank?) if model.tag
30  end
31 end
```

Figura 4.16: Colección Requisición definida en el modelo *Requisition*

```

1 class Product
2   include Mongoid::Document
3   include Mongoid::Paperclip
4   include Mongoid::Paranoia
5   include Mongoid::Slug
6   include Mongoid::Timestamps
7
8   belongs_to :provider
9
10  field :name, type: String
11  field :brand, type: String
12  field :model, type: String
13  field :price, type: BigDecimal
14  field :state, type: String
15  field :description, type: String
16  field :details, type: String
17  field :category, type: Array
18  slug :name, max_length: 24
19
20  has_mongoid_attached_file :image,
21    styles: {
22      original: '1920x1680>',
23      small: '128x128#',
24      thumb: '180x180',
25      medium: '250x250',
26      large: '500x500>'
27    },
28    default_url: 'product/default.jpg'
29
30  scope :search, -> (category) { where(category: category) }
31
32  validates :name, :brand, :model, :price, :state, :description, :category, presence: true
33  validates_attachment_content_type :image, :content_type => /\Aimage\/.*\Z/
34
35  before_validation do |model|
36    model.category.reject!(&:blank?) if model.category
37  end
38 end

```

Figura 4.17: Colección Productos definida en el modelo Products

```

1 class Quotation
2   include Mongoid::Document
3   include Mongoid::Paranoia
4   include Mongoid::Slug
5   include Mongoid::Timestamps
6
7   embeds_many :items, inverse_of: :quotation
8   has_many :comments, dependent: :destroy
9   accepts_nested_attributes_for :items,
10  reject_if: ->(attrs) { attrs['name'].blank? || attrs['quantity'].blank? },
11  allow_destroy: true
12
13  belongs_to :provider
14  belongs_to :requisition
15
16  field :title, type: String
17  field :comentary, type: String
18  field :total, type: String
19  slug :title, max_length: 24
20
21  validates :title, :total, presence: true
22
23 end

```

Figura 4.18: Colección Cotizaciones definida en el modelo Quotation

```

1  class Item
2    include Mongoid::Document
3
4    embedded_in :quotation, inverse_of: :items
5
6    field :name, type: String
7    field :quantity, type: Integer
8    field :select, type: String, default: ""
9
10   validates :name, :quantity, presence: true
11 end

```

Figura 4.19: Colección Elementos de cotización definida en el modelo Item

```

1  class RequisitionTag
2    include Mongoid::Document
3    include Mongoid::Timestamps
4
5    field :name, type: String
6
7    validates :name, presence: true
8    validates :name, length: { minimum: 3 }
9  end

```

Figura 4.20: Colección Categorías definida en el modelo RequisitionTag

```

1  class Comment
2    include Mongoid::Document
3    include Mongoid::Timestamps
4
5    belongs_to :user
6    belongs_to :quotation
7
8    field :body, type: String
9
10   validates :body, presence: true
11 end

```

Figura 4.21: Colección Comentarios definida en el modelo Comment

4.5 Políticas de autorización

Dentro de la estructura de la aplicación, se encuentra una carpeta que contiene archivos planos de Ruby, que definen las políticas de autorización en la aplicación. La carpeta en cuestión se llama *policies* y fue autogenerada por la gema *pundit*, que fue agregada a la aplicación para cumplir con este objetivo (ver figura 4.12).

A continuación, se detallan las políticas de autorización creadas para la aplicación web de este proyecto.

4.5.1 Políticas de requisiciones

En estas políticas se define la autorización que tienen los usuarios para con todos los registros de requisiciones. Consiste básicamente en que cada requisición creada por un usuario requirente es exclusiva para el mismo, es decir, tan solo el usuario dueño de la requisición puede hacer uso de las opciones de modificación o eliminación.

4.5.2 Políticas de cotizaciones

Al igual que las políticas de las requisiciones, solo los usuarios proveedores dueños de una cotización, pueden hacer uso de las opciones de modificación o eliminación de la misma. Esto también aplica para los usuarios requirentes, ninguno puede alterar las cotizaciones realizadas por algún proveedor.

4.5.3 Políticas de productos

Al igual que las políticas anteriores, los productos están limitados en su modificación o eliminación a sus respectivos proveedores creadores. Una diferencia importante, es la que de los productos quedan limitados para su observación entre proveedores, es decir, el producto ingresado por un proveedor solo puede ser visto en detalle por el mismo proveedor.

4.6 Secciones de la aplicación

Las secciones de la aplicación la comprenden las pantallas con la que los usuarios interactúan al momento de acceder a la aplicación. En los puntos a continuación se describirá el proceso que se realiza en cada una las pantallas de la aplicación. Cabe aclarar que las figuras que se mostrarán a continuación, como ejemplos visuales de las pantallas, fueron tomadas durante la ejecución en desarrollo de la aplicación.

4.6.1 Manejo de sesiones

Para el manejo de sesiones, se realiza a través un formulario que pide llenarse con el correo electrónico y contraseña del usuario, que previamente debe haberse registrado a la aplicación. Se cuenta con dos tipos de formularios, uno para los requirentes y otro para los proveedores. Ver figura 4.22

4.6.2 Registro de usuarios

El registro se lleva a cabo a través de un formulario en donde se pide al usuario la información correspondiente según el rol que quiera asumir en la aplicación. Para esto, existen dos tipos de formularios, uno para el registro de requirentes, en donde tan solo se pide información básica de la persona, y otro para proveedores, en donde se solicita, además de la información básica, información legal sobre el mismo. En ambos, puede ser subidas una imagen que identifique a la cuenta por crear. Ver figura 4.23

4.6.3 Requisiciones

Para el registro de requisiciones, se debe contar primero con una cuenta de tipo requirente. En esta sección de la aplicación se presentará un formulario en el que se debe completar un título para la requisición, una descripción sobre lo que se requiere y por último seleccionar una categoría que corresponda a lo solicitado. Ver figura 4.24.

4.6.4 Ingreso de productos

Para el registro de los productos, se debe contar con una cuenta de usuario de tipo proveedor. El objetivo de esta sección es ingresar los productos al sistema que podrían ser cotizados en alguna requisición. Para ingresar un producto se llena un formulario que pide información sobre los detalles del producto, incluyendo una imagen del mismo. Ver figura 4.25.

4.6.5 Cotizaciones

El registro de cotizaciones, que pueden hacerlo únicamente los proveedores, se ubica como sección en cada requisición hecha en el sistema. Los usuarios proveedores pueden entrar en una requisición y proceder a rellenar el formulario de cotizaciones si ellos desean realizar una. El formulario comprende un campo para el título de la cotización, una caja de comentarios para realizar alguna anotación o asesoría, y finalmente un botón para agregar los productos (previamente ingresados) que se quieran cotizar. Ver figura 4.26.

4.6.6 Modificación de registros

A excepción de los comentarios, cada requisición, producto y cotización puede ser editado o eliminado. Para realizarlo, se debe dirigir, iniciada sesión con el correspondiente tipo de usuario, a el registro que se desea modificar y dar clic en 'Editar' o 'Eliminar'. Al dar clic en 'Editar' aparecerá un formulario con los datos del registro, listos para ser modificados a gusto del usuario; por el contrario, al dar clic en 'Eliminar', aparecerá primero un mensaje de advertencia, en el que el usuario decidirá si finalmente elimina o no el registro. Ver figura 4.27.

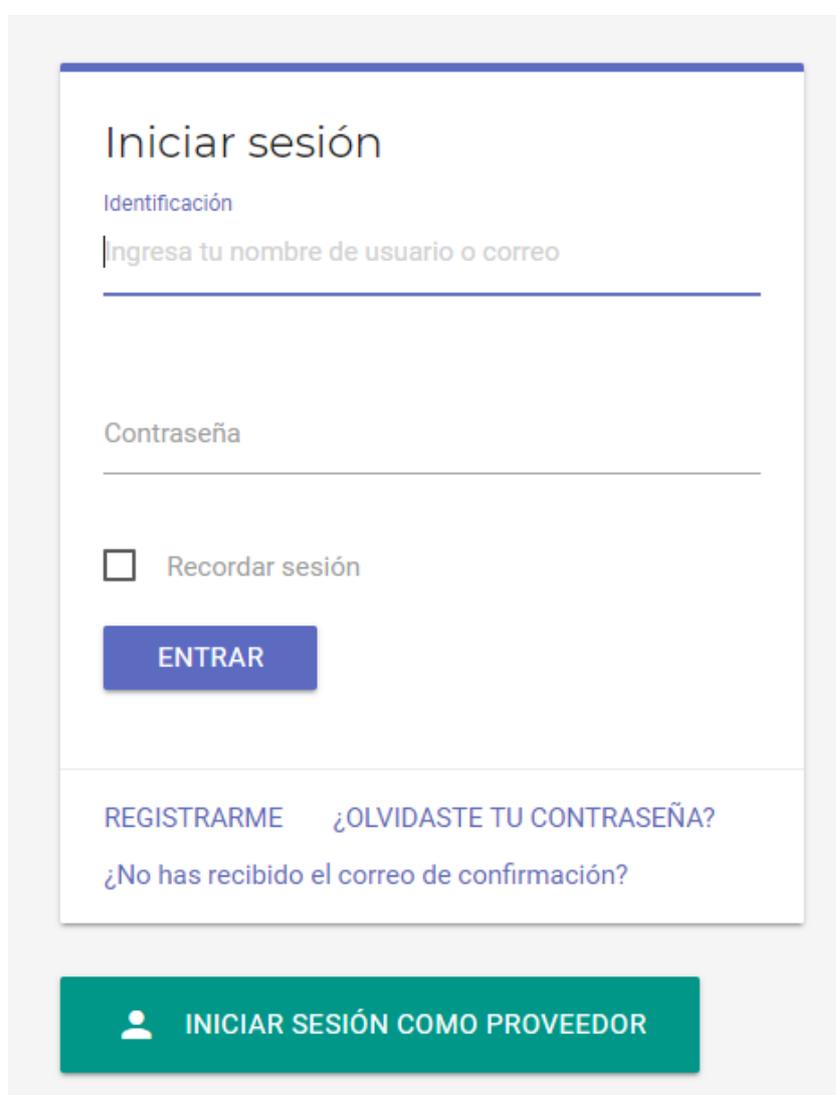
4.6.7 Comentarios

Esta sección solo aparece dentro de la sección de cotizaciones. Permite realizar un comentario sobre la cotización realizada y sirve como medio de comunicación entre el proveedor y el requirente. Tan solo el dueño de la cotización (un proveedor) como el dueño de la requisición en donde se

encuentra la misma (un requirente), pueden realizar comentarios. El formulario de registro de comentario consta de tan solo un campo en donde se puede escribir lo que se desee transmitir. Ver figura 4.28

4.6.8 Panel de control

Sección única y exclusiva del usuario administrador. En esta, el usuario administrador puede controlar el contenido publicado en la aplicación, pudiendo modificar o eliminar cualquier registro si así lo cree conveniente. El propósito de esta sección es la de poder moderar los contenidos publicados, para que no se alejen del propósito de la aplicación. Ver figura 4.29



Iniciar sesión

Identificación

Ingresar tu nombre de usuario o correo

Contraseña

Recordar sesión

ENTRAR

REGISTRARME ¿OLVIDASTE TU CONTRASEÑA?

¿No has recibido el correo de confirmación?

 INICIAR SESIÓN COMO PROVEEDOR

Figura 4.22: Control de sesiones en Cotipro

The image shows two registration forms side-by-side. The left form is titled 'Registrarse' and has the following fields: 'Nombre completo', 'Nombre de usuario', 'Correo electrónico', 'Tu imagen de usuario' (with a dashed box and a plus sign), 'Contraseña' (with a note '(6 caracteres mínimo)'), and 'Repite la contraseña'. At the bottom, there is a blue 'REGISTRAME' button and links for 'INICIAR SESIÓN' and '¿NO HAS RECIBIDO EL CORREO DE CONFIRMACIÓN?'. The right form is also titled 'Registrarse' and has two tabs: 'PASO #1: REGISTRATE' and 'PASO #2: DATOS DE PROVEEDOR'. The 'PASO #2' tab is active and contains fields for 'Nombre proveedor', 'Razón social', 'RUC', 'Correo de contacto', 'Teléfono convencional', 'Teléfono móvil', 'Dirección', 'Logo o imagen del proveedor' (with a dashed box and a plus sign), and 'Breve descripción'. It also has a blue 'REGISTRAME' button, an 'ATRÁS' button, and the same login/confirmation links as the first form.

Figura 4.23: Formularios de registros de requirentes y proveedores, en orden de izquierda a derecha

The image shows a form titled 'Nueva requisición'. It has a 'Título' field, a 'Descripción' field with a rich text editor toolbar (containing icons for undo, redo, bold, italic, underline, link, unlink, x₂, x², bulleted list, numbered list, indent, outdent, link, unlink, and code), and a 'Categorías' field. Below the description field is a blue 'ENVIAR' button with a right-pointing arrow. At the bottom left, there is a link 'MIS REQUISICIONES' with a list icon.

Figura 4.24: Formulario de registro de requisiciones.

Nuevo producto

Nombre

Marca

Modelo

Precio

Estado

Imagen del producto

+

Breve descripción

Detalles adicionales (opcional)

Categorías

ENVIAR >

[MIS PRODUCTOS](#)

Figura 4.25: Formulario de registro de productos.

Realizar cotización

Título

Comentario (opcional)

↶ ↷ **B** *I* U ~~S~~ x_2 x^2 ☰ ☷ ☹ ☺ ✍ 🔄 </>

Escriba algo

0

Cotización

+ AGREGAR PRODUCTO

Total de productos: 0

Total

ENVIAR >

Figura 4.26: Formulario de registro de cotizaciones.

Editando requisición

Título
Otra más de prueba

Descripción

↶ ↷ **B** *I* U ~~S~~ x₂ x²

Aquí otra requisición para probar, en la que se pide lo siguiente:

1. Esto de aquí
2. Esta otra cosa importante de acá
3. Y lo de allá

Saludos

129

Categorías

x Materiales

Recibir cotizaciones

Abierto Cerrar

ENVIAR >

[VER REQUISICIÓN](#)

Figura 4.27: Ejemplo de modificación de registro. Modificando requisición.

Proveedor Dos
proveedordos@contacto.com

● Oferta abierta

Una cotización de prueba

31 de enero de 2018 a las 16:34

Esto es una prueba de cotización, espero que lo que vaya a cotizar sirva.

Además, te realizo algunas recomendaciones:

1. Esto de aquí
2. Lo de acá
3. Etc.

Saludos

COTIZACIÓN:

Varillas corrugadas, Produc, 12cm	3	\$135.60
Porcelanato, Elizabeth, Metro Beige Caixa 1.58m ²	1	\$29.90
		Total: \$165.50

Realizar comentario

Escribe tu comentario aquí...

COMENTAR >

Manuel Sandoya escribe:

Comentario de prueba

Publicado: 22/02/2018

Figura 4.28: Sección de comentarios en una cotización

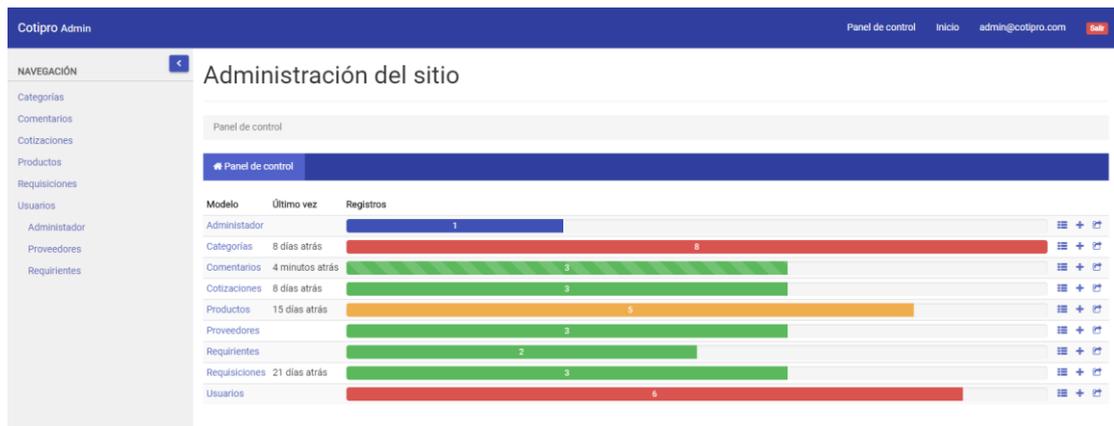


Figura 4.29: Panel de control del usuario Administrador.

4.7 Servicios externos

Para que la aplicación pudiera ejecutar ciertos servicios, como el almacenamiento y muestra de imágenes o el envío de correo electrónico, fue necesaria la utilización de servicios externa. De la misma, un servicio externo fue necesario para alojar la aplicación y que esta esté en línea.

4.7.1 Servicio de mensajería Gmail

Para todo envío de correo electrónico a través de la aplicación, se decidió utilizar el servicio de mensajería Gmail. Perteneciente a Google, este servicio fue utilizado para:

1. Enviar verificaciones de cuenta cuando un usuario envía completo el formulario de registro.



Figura 4.30: Correo de verificación de cuenta de Cotipro.

2. Notificar a los usuarios requirentes cuando hayan recibido una cotización en alguna de sus requisiciones.



Figura 4.31: Correo de notificación de cotización realizada.

4.7.2 Almacenamiento de imágenes en *bucket* S3 de AWS

Para el almacenamiento de imágenes que se suban a la aplicación, como los avatares de usuario o las muestras visuales de un producto, se decidió utilizar un *bucket* S3 de Amazon Web Services (AWS), que ofrece un almacenamiento gratuito de cinco gigabytes por un año (a partir de noviembre de 2017).

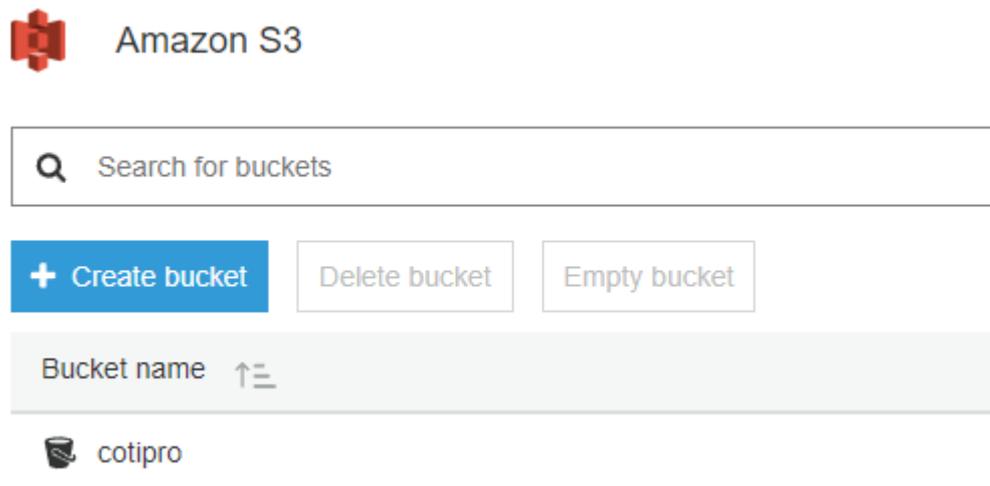


Figura 4.32: Bucket 'cotipro' en S3 de AWS

4.7.3 Plataforma como servicio (PaaS) Heroku

La infraestructura en línea sobre la que está montada la aplicación web se encuentra en Heroku. Heroku es una plataforma como servicio (PaaS según su acrónimo en inglés) de computación en la nube que soporta distintos lenguajes de programación, entre ellos Ruby.

El motivo de su elección es por su facilidad en el manejo de proyectos relacionados con el *framework* Ruby on Rails, y porque abstrae de manera significativa la configuración y montaje de la infraestructura de la aplicación, permitiendo al desarrollador concentrarse en el diseño de la aplicación. Por otro lado, Heroku ofrece servicios gratuitos para el almacenamiento de los proyectos, teniendo como límite 500 megabytes, suficientes para el proyecto en cuestión.

Se debe agregar que, dentro de Heroku, se pueden instalar '*add-ons*' que permiten agregarle funcionalidades a la infraestructura de la aplicación. Es así que, para el motor de base de datos de la aplicación, el cual necesita que sea MongoDB, se agregó el *add-on* mLab, que permite ejecutar una instancia de base de datos MongoDB. Cabe aclarar que este *add-on* también es gratuito y tiene un límite de 600 megabytes para el almacenamiento de registros.

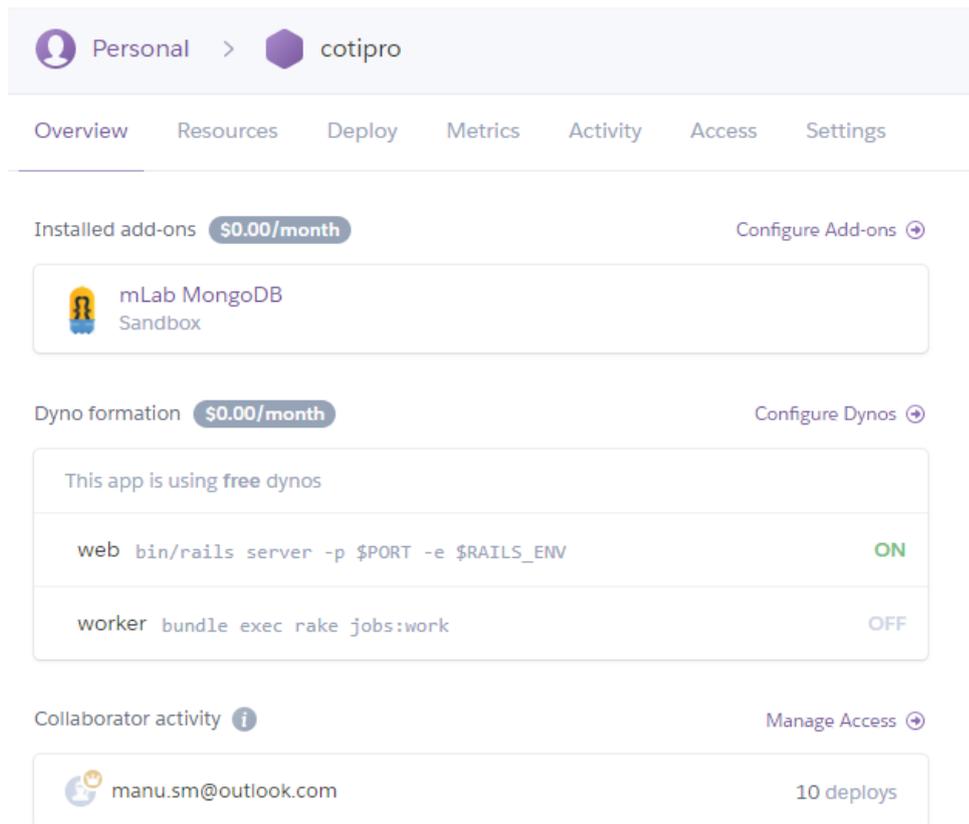


Figura 4.33: Parte del dashboard de Heroku, en donde se encuentra alojada la aplicación

El enlace web hacia la aplicación montada en Heroku es:
<https://www.coti.pro/>.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se detallarán las conclusiones a las que ha llegado esta investigación con respecto al análisis realizado del comportamiento de profesionales y proveedores en el ámbito de la construcción y como la aplicación web propuesta logró mejorar su interacción. Además, se dará recomendaciones con respecto a posibles mejoras en el aplicativo que pudieran hacerse.

5.1 Conclusiones

En el análisis de los datos obtenidos a través de las encuestas realizadas a profesionales y proveedores en el ámbito de la construcción, se evidenció que los profesionales generalmente buscan información sobre productos a través de internet, sin embargo, en la experiencia de muchos de ellos, esta forma no es del todo eficiente, lo que obliga su consulta a sus contactos más cercanos. Algo parecido sucede con los proveedores, si bien saben que sus clientes potenciales están en internet, no existe una forma clara o eficiente de encontrarlos.

Por otro lado, se demostró que el aplicativo web propuesto en este proyecto de titulación es factible porque mejoraría sustancialmente la comunicación entre estos en cuanto realizar búsqueda de información acerca de productos para la construcción o encontrar clientes potenciales, según corresponda.

También, se demostró que la aplicación puede ampliar las fuentes o contactos de los profesionales y proveedores de manera más efectiva, debido a que el encuentro de estos en el aplicativo se basa en el cumplimiento de las necesidades que puedan satisfacerse en una cotización, es decir, un profesional puede conocer a proveedores que antes no en base a las cotizaciones que reciba, mientras que los proveedores pueden conocer a todo cliente potencial que antes no en base a requerimientos más directos y específicos.

Cabe destacar, el aplicativo fue montado de forma exitosa y es accesible en todo el mundo para su utilización, respondiendo a las necesidades de los usuarios que se registren en ella.

5.2 Recomendaciones

Una consideración importante es que la condición actual en la que se encuentra alojado el aplicativo en la plataforma Heroku y la de los servicios externos utilizados, es la de estar siendo utilizados bajo planes de capacidad fija en los mismos, esto significa que hay un límite para el registro de datos (requisiciones, cotizaciones, productos, etc.) y el almacenamiento de imágenes. Ante el futuro escenario de escaso almacenamiento o llegado al límite de registros, escenario que puede ocurrir dentro de 2 o 3 años, se recomienda pasar a planes de capacidad dinámica en los servicios mencionados, que se adapten a las nuevas necesidades, tomando en cuenta la rentabilidad de la aplicación y los recursos disponibles.

En otras consideraciones, la forma en como son dadas las cotizaciones actualmente en el aplicativo dependen de que el proveedor las realice en cada requisición que este vea, sin embargo, este proceso puede ser mejorado con una automatización. Esta automatización se refiere a que, en vez de que los proveedores tengan que realizar las cotizaciones, sería el aplicativo el que se encargaría de realizarlas en base a los productos que el proveedor haya subido a la aplicación y lo que se solicita en cada requisición, reduciendo el trabajo del proveedor y tiempo de espera del profesional. Cabe mencionar que esta automatización debe tener en cuenta todos los posibles controles que deben ser dados al proveedor para que sus cotizaciones estén acorde a sus preferencias.

BIBLIOGRAFÍA

- Alessandro Molina. (2013). *MongoTorino 2013 - BSON Mad Science for fun and profit*. Technology. Recuperado a partir de https://www.slideshare.net/__amol__/mongo-torino-2013-bson-mad-science-for-fun-and-profit
- Barzanallana Asensio, R. M. (2012, noviembre 3). ¿Qué son los lenguajes de marcado? Recuperado 16 de noviembre de 2017, a partir de <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Que-son-lenguajes-marcado.html>
- Cabot Sagrera, J. (2013). *Escaneando la informática: Ingeniería del software*. Barcelona: Editorial UOC. Recuperado a partir de <http://site.ebrary.com/id/10623895>
- Córcoles Tendero, J. E., & Montero Simarro, F. (2014). *Diseño de interfaces web*. Madrid: RA-MA Editorial. Recuperado a partir de <http://public.ebib.com/choice/publicfullrecord.aspx?p=3229167>
- Dimes, T., & Jimenez, M. (2015). *Conceptos Básicos De Scrum: Desarrollo De Software Agile Y Manejo De Proyectos Agile*. Babelcube Inc. Recuperado a partir de <https://www.hoopladigital.com/title/11445161>
- Granados La Paz, R. L. (2014). *Desarrollo de aplicaciones web en el entorno servidor (UF1844)*. Madrid: IC Editorial. Recuperado a partir de <http://public.ebib.com/choice/publicfullrecord.aspx?p=4184020>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. México, D.F.: McGraw-Hill Education.
- López Sanz, M., Vara Mesa, J. M., Verde Marín, J., Sánchez Fúquene, D. M., Jiménez Hernández, J. J., & Castro Martínez, V. (2014). *Desarrollo web en entorno servidor*. Madrid: RA-MA Editorial. Recuperado a partir de <http://site.ebrary.com/id/11046847>
- Mohedano, J., Saiz, J. M., & Salazar Román, P. (2012). *Iniciación a javascript*. Madrid: Ministerio de Educación de España. Recuperado a partir de <http://www.digitaliapublishing.com/a/20926/>
- Muñoz Eslava, V. J. (2012). *HTML, presente y futuro de la web*. Madrid: Bubok Publishing S.L.

- Noguera Otero, F. J., & Riera Terrén, D. (2013). *Programación* (Primera edición). Barcelona: Editorial UOC. Recuperado a partir de <http://site.ebrary.com/id/10853420>
- Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Barcelona: Editorial UOC.
- Vara Mesa, J. M., López Sanz, M., Granada, D., Irrazábal, E., Jiménez Hernández, J. J., & Verde Marín, J. (2014). *Desarrollo web en entorno cliente*. Madrid: RA-MA Editorial. Recuperado a partir de <http://public.ebib.com/choice/publicfullrecord.aspx?p=3229683>
- Zofío Jiménez, J. (2013). *Aplicaciones web*. Madrid: Macmillan Iberia, S.A. Recuperado a partir de <http://site.ebrary.com/id/10820640>

ANEXOS

1. Encuesta: 'Cotizaciones en línea' para profesionales de la construcción.

Cotizaciones en línea

*Obligatorio

¿Por qué medios usted se entera de los lugares donde los productos (materiales de construcción) son más baratos o están en oferta? *

Publicidad en medios escritos o radiales

Contacto o visitas de proveedores

Internet

Otro: _____

Si utiliza internet ¿cuál de estos espacios utiliza más frecuentemente para enterarse de promociones y precios de productos (materiales de construcción)?

Redes sociales

Foros o blogs especializados

Suscripción de correo electrónico

Para la elaboración de presupuestos, los costos de los productos (materiales de la construcción) los obtiene mediante: *

Contacto directo con proveedores

Revistas especializadas

Colegas o gente cercana con experiencia

Otro: _____

Defina el nivel de dificultad que usted tiene para conocer los precios de cada uno de los productos (materiales de construcción) que tiene que adquirir para iniciar un trabajo de construcción. *

	1	2	3	4	5	
Muy fácil	<input type="radio"/>	Muy difícil				

¿Considera usted que sus fuentes son lo suficientemente eficientes para hallar los productos que usted busca? *

- Sí, confío en mis fuentes y no necesito nada más
- No estoy seguro
- No, desearía poder conocer otras fuentes

¿Cree usted que los avances tecnológicos le han ayudado a realizar más eficientemente la búsqueda de los productos que requiere? *

- Sí, me han ayudado mucho
- No lo suficiente
- No, siento que no me han ayudado

¿Utilizaría un sitio web en donde pudiera realizar peticiones de los productos que requiere directamente con los proveedores? *

- Sí, me gustaría
- No lo considero necesario

ATRÁS

ENVIAR

Nunca envíe contraseñas a través de Formularios de Google.

2. Encuesta: 'Cotizaciones en línea para clientes' para proveedores de materiales para la construcción.

Cotizaciones en línea para clientes

*Obligatorio

Encuesta

¿A través de qué medios usted oferta sus productos al consumidor? *

Publicidad

Visita en obras

Internet

En caso de utilizar internet ¿Cuál de estos espacios utiliza/utilizaría más frecuentemente para promocionar sus productos? *

Redes sociales

Mi propia página web

Correo electrónico

¿Qué nivel de dificultad usted tiene para hallar a sus clientes potenciales? *

	1	2	3	4	5	
Muy fácil	<input type="radio"/>	Muy difícil				

¿Por qué? *

Tu respuesta

¿Cómo entrega a sus clientes la cotización de sus productos? *

Por correo electrónico

Personalmente

¿Utilizaría un sitio web que le brindara la oportunidad de ofertar directamente con sus clientes potenciales a través de la realización de cotizaciones virtuales? *

Sí, me gustaría

No lo veo necesario

En ese sitio web ¿Cómo le gustaría presentar sus cotizaciones a sus potenciales clientes? *

En forma pública

En forma privada

ATRÁS

ENVIAR

Nunca envíes contraseñas a través de Formularios de Google.

3. Carta de requerimientos



Guayaquil, 20 de octubre de 2017

Ingenieras,
Beatriz Del Pilar Guerrero Yopez, DIRECTORA
Angela Olivia Yanza Montalvan, COORDINADORA DE TITULACIÓN
Carrera de Ingeniería en Sistemas
Universidad Católica de Santiago de Guayaquil
Presente

De mi consideración:

Por medio de la presente se solicita a ustedes la implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas, el cual debe contar con:

- Módulo de manejo de sesión, para usuarios proveedores y clientes (profesionales)
- Módulo de registro y manejo de requisiciones
- Módulo de registro y manejo de productos
- Módulo de registro y manejo de cotizaciones
- Módulo de comentarios en cotizaciones
- Panel de administración (bajo un usuario administrador)
- Manejo de imágenes (para usuarios y logotipos de proveedores)

Para mejorar los procesos de comunicación entre proveedores y profesionales que se desenvuelven en el área de la construcción, entre los cuales estarán mis actuales y futuros clientes interactuando con mi empresa.

Agradezco su atención.

Atentamente,

GARY RAFAEL GILER PÁRRAGA
MATERIALES ELÉCTRICOS GARY GILER
Gerente General

C.I.: 1308157351

Calle Rocafuerte 1006 y 9 de Octubre Teléfono 2650756 – 0980008811 Portoviejo - Ecuador

4. Carta de conformidad



Guayaquil, 21 de febrero de 2018

Ingeniera,
BEATRIZ DEL PILAR GUERRERO YÉPEZ
Directora
Carrera de Ingeniería en Sistemas Computacionales
Ciudad

Des mis consideraciones,

Yo, GARY RAFAEL GILER PÁRRAGA, informo a usted que estoy plenamente conforme con la aplicación web Cotipro que fue desarrollada en el proyecto de titulación: "Implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas", elaborado por el estudiante Manuel Alejandro Sandoya Mendoza, cumpliendo con todos los requerimientos planteados.

Atentamente

GARY RAFAEL GILER PÁRRAGA
MATERIALES ELÉCTRICOS GARY GILER
Gerente General

Calle Rocafuerte 1006 y 9 de Octubre Teléfono 2650756 – 0980008811 Portoviejo - Ecuador



DECLARACIÓN Y AUTORIZACIÓN

Yo, **Sandoza Mendoza Manuel Alejandro**, con C.C: # 0930052253 autor/a del trabajo de titulación: **Implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas**, previo a la obtención del título de **Ingeniero en Sistemas Computacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 9 de marzo de 2018

f. 

Nombre: **Sandoza Mendoza, Manuel Alejandro**

C.C: **0930052253**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Implementación de un aplicativo web para la requisición, asesoramiento y cotización de productos para la construcción de obras civiles y arquitectónicas.		
AUTOR(ES)	Sandoya Mendoza, Manuel Alejandro		
REVISOR(ES)/TUTOR(ES)	Pesantes Mendez, Jorge Salvador		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería en Sistemas Computacionales		
TÍTULO OBTENIDO:	Ingeniero en Sistemas Computacionales		
FECHA DE PUBLICACIÓN:	9 de marzo de 2018	No. DE PÁGINAS:	87
ÁREAS TEMÁTICAS:	Desarrollo de software, Tecnologías y sistemas productivos		
PALABRAS CLAVES/KEYWORDS:	APLICATIVO WEB; REQUISICIÓN; COTIZACIÓN; CONSTRUCCIÓN; PROVEEDORES		
RESUMEN/ABSTRACT:	<p>El presente proyecto de titulación consiste en el análisis de las formas en cómo profesionales y proveedores en el ámbito de la construcción se desenvuelven en el proceso de realizar u obtener una cotización, y cómo la implementación de un aplicativo en la web, basado en los principios de la web 2.0, puede mejorar el mismo. El análisis se enfoca en las actuales limitaciones que tanto profesionales y proveedores de la construcción tienen al momento de conseguir información sobre un producto o encontrar clientes potenciales, según corresponda. Así mismo, en el presente documento se explica cómo se desarrolló el aplicativo web que pretende mejorar este proceso, detallando las diferentes tecnologías que se utilizaron para alcanzar los objetivos planteados. Por último, se muestran las conclusiones alcanzadas en este proyecto y se dan recomendaciones para realizar iteraciones de mejoras en el aplicativo web propuesto.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTORES:	Teléfono: +593978751471	E-mail: manuel.sandoya95@gmail.com	
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::	Nombre: Yanza Montalvan, Ángela Olivia		
	Teléfono: +593983035702		
	E-mail: angela.yanza@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			

Manuel Sandoya