



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

CREACIÓN DE UNA PLATAFORMA PARA LA DIFUSIÓN DE  
PRODUCTOS ARTESANALES DEL ECUADOR UTILIZANDO A METEOR  
COMO FRAMEWORK

AUTOR

Alberto Sebastián Tatés Puetate

AÑO

2017



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

CREACIÓN DE UNA PLATAFORMA PARA LA DIFUSIÓN DE PRODUCTOS  
ARTESANALES DEL ECUADOR UTILIZANDO A METEOR COMO  
FRAMEWORK

Trabajo de Titulación presentado en conformidad a los requisitos establecidos  
para optar por el título de  
Ingeniero en Sistemas de Computación e Informática

Profesor Guía  
MSc. Santiago Ramiro Villarreal Narváz

Autor  
Alberto Sebastián Tatés Puetate

Año  
2017

## **DECLARACIÓN DEL PROFESOR GUIA**

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

---

Santiago Ramiro Villarreal Narvárez

Master Science Technologies, Sante A Finalite Recherche Mention

Informatique Specialite Web Intelligence

C.I: 1713980074

## **DECLARACIÓN DEL PROFESOR CORRECTOR**

Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación

---

Bernarda Cecibel Sandoval Romo  
Master en Ciencias de la Computación  
C.I: 1709974453

## **DECLARACION DE AUTORIA DEL ESTUDIANTE**

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen a los derechos de autor vigentes.

---

Alberto Sebastian Tatés Puetate

C.I: 0401491287

## **AGRADECIMIENTOS**

A la serie de acontecimientos que me llevaron a elegir la carrera de ingeniería en sistemas computacionales e informática.

A mi madre y a mi padre por su apoyo incondicional.

A todos los profesores que desinteresadamente enriquecieron mi perfil personal y profesional.

A todas las personas que estuvieron en este trayecto.

## **DEDICATORIA**

Para cada uno de los profesores que fueron parte de mi formación profesional y personal de la Universidad de las Américas.

Para Tulcán mi tierra natal.

Para todos los estudiantes que están buscando culminar su carrera universitaria.

Para la vida y todas sus sorpresas.

## RESUMEN

El sector artesanal es muy representativo en Ecuador con un promedio de 48 mil artesanos que generan 980 mil empleos según el IESS en 2016. Pese a los diferentes medios de difusión, existe una necesidad presente de un medio unificado que sirva para potenciales clientes como un catálogo global del trabajo del sector artesanal. Para solventar esta necesidad, se ha utilizado medios digitales en la Web como redes sociales o páginas de compra y venta de artículos. Estos medios ayudan al sector a difundir su trabajo, pero no están centrados en el problema de un medio unificado para productos y servicios artesanales.

El presente trabajo de titulación tiene como finalidad la entrega de una aplicación web que sirva como plataforma centralizada para ofrecer servicios y productos artesanales, además que facilite la comunicación entre un potencial cliente y un proveedor. Para alcanzar el objetivo mencionado y de acuerdo a los requerimientos obtenidos con dos trabajadores artesanales, se elige la metodología de desarrollo ágil SCRUM por la adaptabilidad a cambios para una mejora continua del proyecto y se elige a Meteor como *framework* para construir la aplicación basada en sus características que suplen las necesidades técnicas surgentes del aplicativo.

## ABSTRACT

The craftwork is representative for Ecuador with 48 thousand workers. Despite different broadcast media, there is present need for a unified way to give a catalog of craftwork products or services to potential clients. To solve this need, some digital media on Internet like social networks or buy and sell websites have been used, even if this media helps craftsmen to spread they work, they don't focus on the need for a centralized platform .

This project has as purpose to build up a web app that works as a centralized platform to offer craft services and products also making the communication between a potential client and craftsmen easier. In order to reach this goal with the craftsmen requirements, SCRUM as an agile methodology for software development has been chosen because it deals with changes and gives continuous improvement. Meteor as framework to build the web application was also chosen because it supplies the technical emerging needs.

## ÍNDICE

1. CAPÍTULO I. INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.1.1 Artesanos.....	1
1.3 Justificación.....	2
1.4 Objetivo General.....	3
1.5 Objetivos específicos.....	3
2. CAPÍTULO II. MARCO TEÓRICO.....	4
2.1 Aplicación web.....	4
2.1.1 Definición.....	4
2.1.2 CMS.....	5
2.1.3 MVC.....	5
2.2 Meteor.....	7
2.2.1 Definición.....	7
2.2.2 Herramienta de línea de comandos.....	8
2.2.3 Servidor.....	8
2.2.4 Capa de Comunicación.....	8
2.2.5 Buscador.....	9
2.2.6 Cordova.....	9
2.2.7 Los paquetes.....	10
2.3 Comparación de un MVC con Meteor.....	10
2.4 Desarrollo ágil de software.....	11
2.5 SCRUM.....	12
2.5.1 Definición.....	12
2.5.2 Proceso.....	13
2.5.3 Roles SCRUM.....	14
2.5.4 PivotalTraker.....	15
2.6 Materialize.....	15
2.7 Heroku.....	15
2.8 Amazon Simple Storage Service (Amazon S3).....	15
3.Capítulo III. Desarrollo de la Aplicación.....	16

3.1 Situación Actual.....	16
3.2 Aplicación Propuesta.....	17
3.1 Planificación o Sprint 0.....	18
3.2 Historia de Usuario uno.....	21
3.3 Historia de usuario dos.....	24
3.4 Historia de usuario tres.....	25
3.5 Historia de usuario cuatro.....	27
3.6 Historia de usuario cinco.....	28
3.7 Historia de usuario seis.....	31
4. Capítulo IV. Resultados.....	34
5. Conclusiones y Recomendaciones.....	36
5.1 Conclusiones.....	36
5.2 Recomendaciones.....	37
Referencias.....	39
ANEXOS.....	41

## INDICE DE TABLAS

Tabla 1. <i>Comparación de cómo crear una aplicación Meteor con Drupal</i> .....	10
Tabla 2. <i>Índices de los procesos de promoción de productos artesanales mediante los medios de Facebook y WordPress</i> .....	17
Tabla 3. <i>Responsables proceso SCRUM</i> .....	17
Tabla 4. <i>Historias de usuario de acuerdo a su valor y a los puntos asignados</i>	19
Tabla 5. <i>Product Backlog</i> .....	20
Tabla 6. <i>Desarrollo de las tareas de la historia uno en cada Sprint utilizado</i> ...	23
Tabla 7. <i>Desarrollo de las tareas de la historia de usuario tres en cada Sprint utilizado</i> .....	26
Tabla 8. <i>Desarrollo de las tareas de la historia de usuario cinco en cada Sprint utilizado</i> .....	29
Tabla 9. <i>Desarrollo de las tareas de la historia de usuario seis en cada Sprint utilizado</i> .....	32
Tabla 10. <i>Resultados</i> .....	34

## INDICE DE FIGURAS

<i>Figura 1.</i> Representación de la arquitectura de una aplicación web. ....	5
<i>Figura 2.</i> Diagrama de Modelo Vista Control.....	6
<i>Figura 3.</i> Revisión de la plataforma Meteor.....	7
<i>Figura 4.</i> Diagrama de Proceso SCRUM.....	14
<i>Figura 5.</i> Backlog de las historias de usuario en la herramienta PivotalTraker .....	21
<i>Figura 6.</i> Formulario de registro, entrega de la historia de usuario uno .....	23
<i>Figura 7.</i> Formulario de información de usuario, entrega de historia de usuario uno.....	24
<i>Figura 8.</i> Página de perfil de un artesano, entrega de historia de usuario dos.....	25
<i>Figura 9.</i> Elemento de mensajería instantánea, entrega historia de usuario tres .....	27
<i>Figura 10.</i> Bandeja de mensajes de un usuario, entrega de la historia de usuario cuatro.....	28
<i>Figura 11.</i> Formulario de información general de una publicación, entrega de la historia de usuario cinco .....	30
<i>Figura 12.</i> Vista de las publicaciones de un usuario, entrega de la historia de usuario cinco .....	30
<i>Figura 13.</i> Resultados de búsqueda con la palabra “cuenca”, entrega de la historia de usuario seis.....	32
<i>Figura 14.</i> Página de un producto o servicio, entrega de la historia de usuario ocho.....	33

## **1. CAPÍTULO I. INTRODUCCIÓN**

### **1.1 Antecedentes**

#### **1.1.1 Artesanos**

En el Ecuador, el campo artesanal se considera un sector representativo con un promedio de 48 mil artesanos que generan 980 mil empleos (IESS, 2016).

La Ley de defensa del Artesano señala que los artesanos son aquellos que practican una actividad eminentemente artesanal con predominio de actividad manual, dirigida personalmente por el artesano; los requisitos para que una persona sea considerada como artesano son: tener máximo un taller y un establecimiento de comercialización; contar con un máximo de 15 operarios y cinco aprendices.

Los productos y servicios artesanales son preferidos por la población por el referente de tradición, cantidad y calidad a la hora de elaborar y repartir el producto final.

En el país, según el secretario a cargo de información en la Junta Nacional de Defensa del Artesano (JNDA) ubicada en la ciudad de Quito, los productos y servicios artesanales son difundidos usando los medios de comunicación de radio, periódico y páginas de compra y venta en internet, sin embargo para la mayoría de artesanos la inversión en propaganda no resulta económicamente conveniente, menos si es una organización con poco tiempo en el mercado.

En los últimos años seis de cada 10 productores artesanos prefieren las redes sociales para difundir sus productos porque alcanzan una buena difusión entre la población.

El secretario de información de JNDA manifestó también que uno de los principales inconvenientes a la hora de ofrecer un producto por internet es la falta de comunicación para el comprador en cuanto a las especificaciones de lo que se ofrece.

## 1.2 Alcance

El alcance final del proyecto de titulación, será una aplicación web creada con el *framework* Meteor que estará disponible en Internet mediante el nombre de dominio [www.craft-ec.com](http://www.craft-ec.com) alojado en el servidor de *hosting* Heroku.

La aplicación cumplirá con los siguientes requerimientos funcionales:

- Permitir a un usuario registrarse en la aplicación por medio de un correo y contraseña o utilizando Facebook.
- Permitir a un usuario ingresar la información de su trabajo artesanal y publicar productos o servicios.
- Permitir buscar y filtrar los resultados de los productos o servicios de acuerdo a una palabra clave.
- Permitir acceder a la información de un producto o servicio seleccionado, y un enlace hacia el perfil del vendedor.
- Permitir acceder a la información de un artesano y todos los productos y servicios que este tenga.
- Permitir iniciar una sesión de chat con el usuario artesano desde su perfil.
- Permitir ver la bandeja de mensajes donde se encuentren todas las conversaciones realizadas.

El proyecto sin embargo no alcanzará requerimientos como:

- Realizar transacciones monetarias.
- Implementar estrategias de *Search Engine Optimization* (SEO).
- Implementar herramientas de análisis como Google Analytics.
- Implementar páginas de información de la aplicación o de ayuda.

## 1.3 Justificación

Según la información obtenida de la JNDA se hace visible la necesidad existente de un método de difusión de productos o servicios artesanales donde

no se presenten limitaciones para realizar publicaciones y que exista una comunicación instantánea entre cliente y proveedor.

La aplicación web propuesta logra un sitio unificado donde el sector artesanal tiene una herramienta para publicar toda su información y hacerla visible junto al trabajo de proveedores que se encuentran en un campo afín. Además es una herramienta que facilita a las personas que están buscando por un servicio o producto artesanal y requieran comunicación inmediata para que sus preguntas sean solventadas.

La plataforma se convierte en una red de artesanos que maximiza el alcance del trabajo de este sector del país.

#### **1.4 Objetivo General**

Crear una aplicación web de tipo catálogo de productos que permita a los productores artesanos del país publicar su trabajo para que puedan ser contactados mediante su información personal o la bandeja de mensajería instantánea de la aplicación construida en el *framework* de tiempo real Meteor.

#### **1.5 Objetivos específicos**

1. Recolectar y analizar los requerimientos a partir de las necesidades de trabajadores artesanos para proponer una solución.
2. Usar SCRUM como metodología de desarrollo para diseñar, construir y desplegar los requerimientos obtenidos convertidos en historias de usuario y categorizados por sus puntos de complejidad.
3. Implementar cambios de acuerdo a la retroalimentación de los usuarios en base a las entregas de cada Sprint para tener un producto final mejorado que esté disponible en Internet.

## 2. CAPÍTULO II. MARCO TEÓRICO

### 2.1 Aplicación web

#### 2.1.1 Definición

En el principio de la informática, la computación se basaba en software que era ejecutado y manejado desde un mismo centro de cómputo donde se encontraba la interfaz con el usuario. Esto no hacía escalable una aplicación puesto que se requiere una actualización global en cada puesto de trabajo de la aplicación para agregar nuevas características, esto añade un gran costo de soporte técnico.

Una aplicación web es un programa informático cliente-servidor, es decir un programa al cual se accede remotamente, donde la interfaz del usuario es un navegador de Internet que realiza las peticiones hacia un servidor a través de la red, dando la apariencia de que el software corre localmente. Las aplicaciones web deben ser escritas en lenguajes soportados por los navegadores. Entonces los componentes básicos de una aplicación web como lo muestra la Figura 1, son el navegador web, el servidor web y la base de datos de la aplicación.

Las aplicaciones web tienen como cliente ligero al navegador de Internet lo que genera una ventaja a la hora de mantener y actualizar la aplicación sin repercusión en el trabajo de los usuarios.

Para acceder a una aplicación web, el software debe estar instalado en un ordenador con acceso a Internet, a partir de aquí se lo alcanzará mediante su dirección *Internet Protocol* (IP) o su nombre de dominio. (Ramos y Ramos, 2015, p. 28)



Figura 1. Representación de la arquitectura de una aplicación web.  
Tomado de bespokesoftwaredevelopment, 2017

### 2.1.2 CMS

Un Sistema de Gestión de Contenidos (Content Management System) es un software que ayuda a construir una estructura de base (*framework*) para el manejo de contenido de aplicaciones o páginas web, por parte de los usuarios que estén gestionando el sistema.

Un CMS tiene una interfaz que controla todos los elementos del sitio web como las bases de datos, los elementos de interfaz gráfica (GUI) de uno o varios usuarios que use la plataforma, el contenido textual y gráfico de la página. El CMS da una amplia apertura para manejar de manera independiente contenido y el diseño, sin embargo como todo programa debe ser instalado y configurado para comenzar a usar. El CMS permite en cualquier momento realizar cambios al diseño del sitio, sin embargo en cuanto a nuevas características muchas veces no pueden ser 100% personalizables. Uno de los CMS más populares es Drupal con más de 106 mil usuarios a nivel mundial. (Margaret Rouse, 2016)

### 2.1.3 MVC

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software, que busca crear programas computacionales que separen los datos y la lógica de

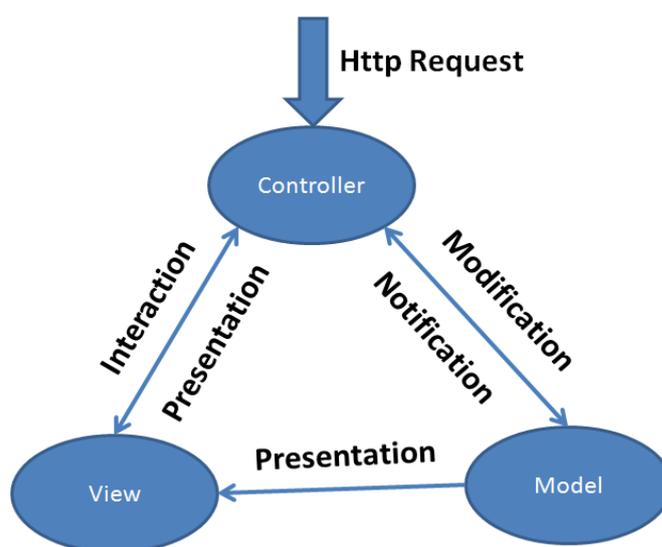
negocio de la interfaz de usuario y manejar cómo estos tres elementos se comunican jerárquicamente. MVC se basa en la construcción de tres componentes como muestra la Figura 2, que son el modelo, la vista y el controlador.

El modelo representa la esquematización de un objeto del negocio que será almacenado en la base de datos.

El controlador está conformado por una serie de funciones siguiendo la lógica del negocio para transformar los datos y guardarlos en el respectivo lugar de almacenamiento dependiendo del modelo.

La vista es la parte de la arquitectura dirigida al usuario con todos los controles para que este interactúe con las capas inferiores.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. (Apple, 2015)



**ASP.Net MVC Design Pattern**

*Figura 2.* Diagrama de Modelo Vista Control.

Tomado de developerin, 2017

## 2.2 Meteor

### 2.2.1 Definición

Meteor.js es una plataforma de desarrollo cohesiva, una colección de librerías y paquetes que son unidos de manera muy ordenada para hacer el desarrollo web más fácil. Meteor se construye sobre ideas de *frameworks* y librerías previas que ofrecen una manera fácil para comenzar una aplicación prototipo, pero da las herramientas y flexibilidad para construir aplicaciones completas y robustas. Hay muchas librerías que el grupo de desarrollo de Meteor a construido específicamente para un *front-end* reactivo. Múltiples componentes hacen posible que Meteor sea la aplicación reactiva como muestra la figura Figura 3. (Owens, 2016)

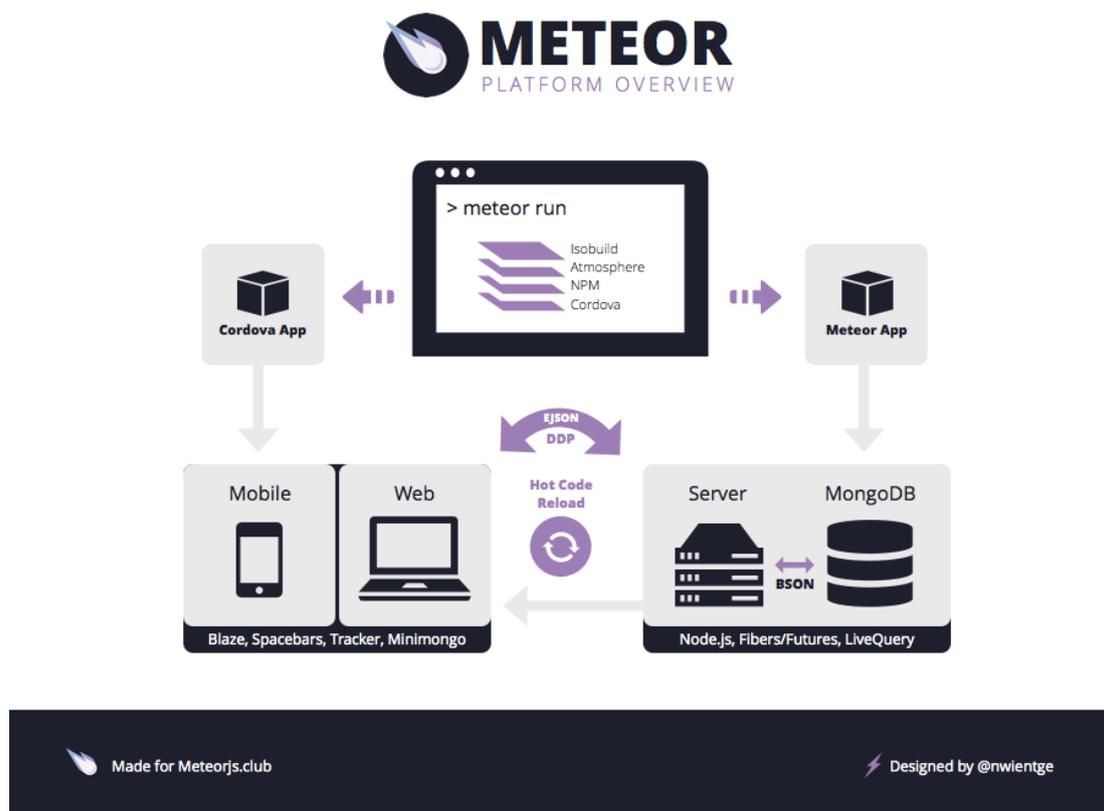


Figura 3. Revisión de la plataforma Meteor.  
Tomado de Owens, 2017

### 2.2.2 Herramienta de línea de comandos

Meteor tiene una herramienta de línea de comandos, esta actúa como desencadenador, esta línea de comandos contiene soporte para compiladores como Coffescript o Less. Esta herramienta contiene también un sistema de paquetes embebido denominado Isobuild.

Isobuild es un sistema de paquetes isomórfico que permite fácilmente instalar paquetes a través de Atmosphere, el manejador de paquetes de Node.js NPM y *plugins* de Cordova.

### 2.2.3 Servidor

El servidor de Meteor es una aplicación de Node.js construida con ciertas librerías para hacer que la comunicación pase sobre Distributed Data Protocol (DDP) y la Javascript Object Notation (EJSON) hacia el *front-end*. Los componentes del servidor son:

- Node.js - Servidor en JavaScript.
- Connect - Librería para generar salidas http desde la aplicación.
- Database Driver (Mongo) - Una interfaz de conexión simple con datos de MongoDB.
- Livequery - Una librería para el envío de información Mongo de manera reactiva.
- Fibers/Futures - Una librería de envoltura para Node.js que hace a la aplicación síncrona para reducir las funciones de *callback*.

### 2.2.4 Capa de Comunicación

La capa de comunicación es la encargada de poner al cliente y el servidor juntos. EJSON es usado para serializar y deserializar la información moviéndose por la red a través de DDP

- DDP - Un protocolo para enviar información por encima de *websockets*.

También llamado “Rest para *websockets*”.

- EJSON - Una extensión de JSON que soporta la serialización de más tipos de datos como fechas o binario.

### 2.2.5 Buscador

El buscador como parte de Meteor es enviado sobre *Hypertext Markup Language* (HTML) mínimo y JavaScript que carga el ambiente. Este utiliza JQuery y Underscore.js para el funcionamiento *front-end*, mientras el servidor es síncrono, el buscador y Javascript son asíncronos por naturaleza. A continuación las librerías que ayudan a que el cliente tenga naturaleza reactiva.

- Tracker - La columna vertebral del *front-end* reactivo. Este es el componente que permite un flujo de datos reactivo en el cliente.
- Spacebars - Una derivación de Handlebars para permitir la reactividad que da la sintaxis para el manejo de *front-end*.
- Blaze - Una librería reactiva construida para unir a Tracker y Spacebars para crear interfaces de usuario que se actualizan en tiempo real.
- Minimongo - Una librería de mongo del lado del cliente que sincroniza los datos sobre DDP y permite al cliente consumir reactivamente información de mongo.
- Session - Una librería para soportar variables de estado de Interfaz de Usuario (UI) reactivas.

### 2.2.6 Cordova

La integración con Cordova es construida en la herramienta de línea de comandos para compilar la aplicación Meteor en una aplicación Cordova, entonces la recarga de código en caliente trae y actualiza un nuevo código para ponerlo en caché del dispositivo móvil corriendo la aplicación y cargarlo para cuando la aplicación reinicie.

La interfaz de Cordova está construida con las mismas librerías del buscador

descritas anteriormente.

### 2.2.7 Los paquetes

ISOBuild, la plataforma de paquetes que se puede agregar a Meteor. Este sistema de paquetes tiene múltiples opciones base para ser agregadas fácilmente como paquetes para autenticación con terceros servicios.

Estos paquetes son pequeñas librerías construidas para hacer el desarrollo web incluso más rápido. Existen múltiples paquetes para reusar y evitar construir software desde el inicio.

### 2.3 Comparación de un MVC con Meteor

El uso de Meteor como framework se justifica en cuanto a la facilidad de construir una aplicación de tipo catálogo web y los procesos de configuración de los componentes necesarios como el servidor de base de datos y la comunicación cliente-servidor en comparación a CMS convencionales como Drupal. Comenzando con la configuración del entorno, en la Tabla 1 se muestran los pasos mínimos para iniciarse con una aplicación en Drupal comparado con Meteor.

Tabla 1

*Comparación de cómo crear una aplicación Meteor con Drupal*

	<b>Meteor</b>	<b>Drupal</b>
<b>Instalación del entorno</b>	<ol style="list-style-type: none"> <li>1. Descargar Meteor</li> <li>2. Instalar</li> <li>3. Empezar a desarrollar</li> </ol>	<ol style="list-style-type: none"> <li>1. Descargar Drupal</li> <li>2. Crear la base de datos</li> <li>3. Crear settings.php y el directorio de archivos</li> </ol>

		<ol style="list-style-type: none"> <li>4. Correr el script de instalación</li> <li>5. Empezar a desarrollar</li> </ol>
<b>Creación de una nueva aplicación</b>	<ol style="list-style-type: none"> <li>1. Ejecutar en línea de comandos: <i>meteor create app [nombre]</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Crear la aplicación en el servicio de Drupal</li> <li>2. Llenar la información necesaria de la aplicación</li> </ol>
<b>Instalación de paquetes</b>	<ol style="list-style-type: none"> <li>1. Ejecutar en línea de comandos: <i>meteor add [nombre del paquete]</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Descargar el paquete o módulo.</li> <li>2. Agregarlo al proyecto</li> </ol>

Drupal es manejado y configurado en su mayor parte por su consola de administración, mientras que para Meteor todo el desarrollo se hace por un editor de texto escribiendo código en lenguaje JavaScript esto lo hace 100% personalizable.

Meteor cuenta con 11919 paquetes agregados por la comunidad (Meteor, 2017) con funcionalidades útiles para agilizar el desarrollo

También se justifica el uso de este *framework* de tiempo real para la mensajería instantánea y la información que se necesita que los usuarios la vean actualizada en tiempo real.

## 2.4 Desarrollo ágil de software

El desarrollo ágil de software consta de una visión o guía acerca de pautas que tomar en los proyectos de software, a partir de métodos de ingeniería de software que se basan en desarrollos iterativos e incrementales, donde tanto el proyecto como sus requisitos y posibles soluciones van emergiendo según el

tiempo de vida del proyecto.

Los proyectos en desarrollo ágil se caracterizan por tener grupos auto organizados y multidisciplinarios.

En cuanto a las entregas incrementales, para cada una de las retribuciones se debe realizar el respectivo procedimiento de: planificación, análisis de requisitos, diseño, desarrollo, pruebas y documentación. Enfatizando que las entregas son software funcional útil para el cliente al contrario de los métodos de desarrollo convencionales que buscaban entregas y una cobertura total desde el inicio del proceso

Los métodos ágiles marcan mayor importancia a las comunicaciones tanto con el cliente como con el equipo de manera presencial en vez de respaldos en la documentación.

En cuanto a los equipos de desarrollo ágil, la mayoría están ubicados en oficinas unificadas para interacciones inmediatas y donde tengan a su disposición herramientas de revisión, escritores de documentación, manejadores de iteración y directores del proyecto. (Manchón, 2016)

## **2.5 SCRUM**

### **2.5.1 Definición**

SCRUM es un proceso donde se aplican de forma periódica un grupo de buenas prácticas para trabajar efectivamente en equipo con el fin de sacar el mejor resultado de un proyecto, estas buenas prácticas se sustentan entre sí. SCRUM nace de un estudio en la manera de cómo trabajar en equipos muy productivos.

SCRUM realiza entregas incrementales regularmente, estas entregas se priorizan por el beneficio que aportan al cliente del proyecto. SCRUM puede ser bien utilizado para proyectos de distinta naturaleza, en especial los que tienen mucho riesgo de cambio.

Las ejecuciones del proyecto según SCRUM se denominan Sprints que normalmente constan de iteraciones de dos hasta cuatro semanas. Cada una de estas iteraciones parte de la lista priorizada de objetivos, requerimientos o historias de usuario a partir del valor que aportan hacia el proyecto completo. (Proyectos Ágiles, s.f)

### **2.5.2 Proceso**

#### **1. Planificación de la iteración**

Al respecto de la lista de requisitos el equipo despeja dudas con el cliente y se seleccionan los requisitos que se entregarán al final de la iteración planificada. Cuando se hayan seleccionado los requisitos y priorizado según el valor que agrega al producto final para el cliente, el equipo elabora la lista de tareas necesarias para completar el sprint junto con la estimación del esfuerzo y la auto-asignación a nivel del equipo.

#### **2. Ejecución de la iteración**

Normalmente el equipo de SCRUM realiza una reunión de sincronización para que cada miembro vea el avance de sus semejantes, así mismo revisen complicaciones y dependencias entre las tareas para realizar adaptaciones necesarias, para esta etapa las tareas han pasado a formar parte del Product Backlog para su control.

En esta etapa, el facilitador o SCRUM Master se encarga de que el equipo pueda cumplir con su objetivo y que la productividad se mantenga.

#### **3. Inspección y adaptación**

Esta etapa consta de la reunión para revisar la iteración, donde primero se realiza una demostración de los requisitos completados en función de cómo se desenvuelve el sprint, una vez recibidos los avances son aceptados o declinados por el cliente para una nueva entrega, la retroalimentación del cliente es indispensable para SCRUM.

La Figura 4 muestra una gráfica de los procesos SCRUM mencionados y los miembros del equipo responsables.

### 2.5.3 Roles SCRUM

El equipo SCRUM debe estar enfocado en construir software de calidad, centrándose en las características que el producto debe tener y sobrepasar obstáculos que se presenten en el camino, los roles SCRUM son los siguientes:

1. *SCRUM Master*. Líder del grupo que guía para que las reglas y procesos de la metodología sean cumplidos, ayuda a reducir los impedimentos.
2. *Product Owner*. Es un representante del cliente, es decir se centra en la parte de negocio y tiene el objetivo de que el producto que se entrega supera las expectativas. Es el cargado de traducir las necesidades del cliente a forma de historias para incorporarlas en el Product Backlog y cambia las prioridades según las necesidades emergentes.
3. *Team*: Grupo de profesionales con conocimientos técnicos necesarios que desarrollan el proyecto de manera conjunta cumpliendo múltiples disciplinas para llevar a cabo las historias.

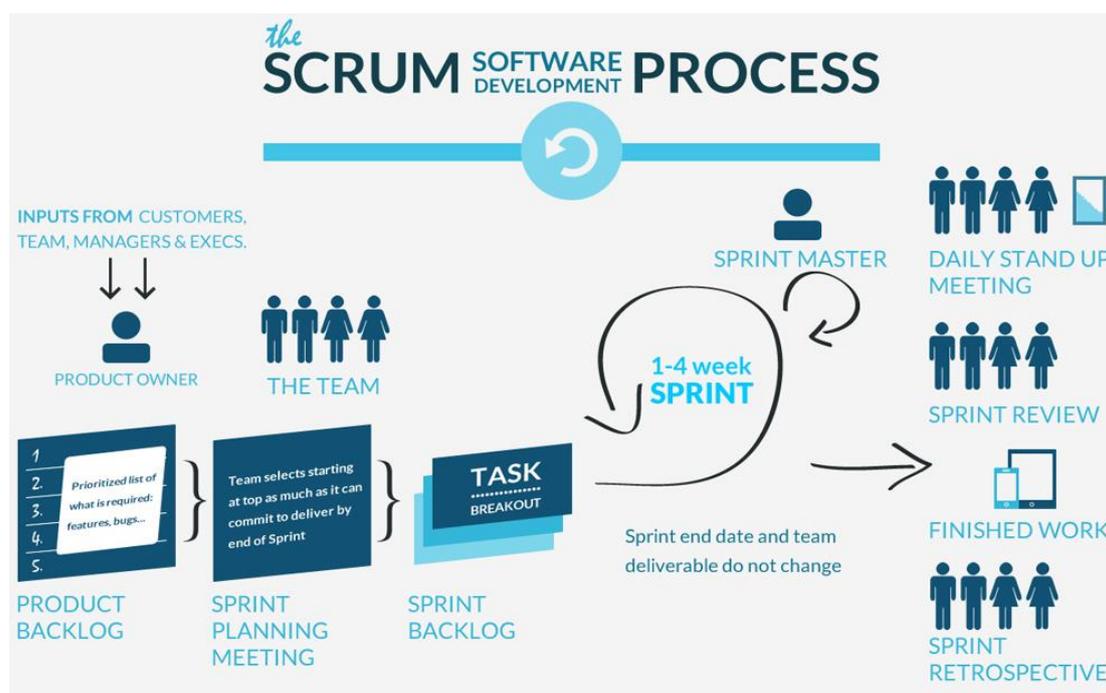


Figura 4. Diagrama de Proceso SCRUM.  
Tomado de maxxor, 2017

#### **2.5.4 PivotalTraker**

Es una herramienta de planificación de proyectos tipo SCRUM que ayuda a los equipos de desarrollo de software a generar expectativas acerca de cuándo un trabajo puede ser completado basado en el desempeño actual. Tracker visualiza el proyecto en forma de historias como cartas moviéndose a través de todo el flujo de trabajo haciendo que los proyectos se desglosen en pedazos manejables y dándoles alcances en cuanto a la entrega, según como el equipo prioriza y estima las historias, el programa las divide en iteraciones futuras. (PivotalTraker, 2016)

#### **2.6 Materialize**

Es un *framework* moderno para *front-end* basado en *Material Design*.

*Material Design* es un lenguaje de diseño creado por Google que combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología. El objetivo de Google consiste desarrollar un sistema de diseño que permite una experiencia de usuario unificada a través de todos sus productos en cualquier plataforma. (Materialize, 2017)

#### **2.7 Heroku**

Heroku es una plataforma en la nube que deja a las compañías construir, entregar, monitorear y escalar en las aplicaciones.

Heroku permite personalizar un servidor con las capacidades necesarias de acuerdo a nuestra aplicación y escalar según sea la necesidad.

Gracias a las plantillas para configuración, el proceso de instalar nuestra aplicación sobre un servidor se reduce en cuanto a costo y permite enfocarse en el desarrollo. (Heroku, 2017)

#### **2.8 Amazon Simple Storage Service (Amazon S3)**

Es una característica de Amazon Web Services que permite almacenar objetos

mediante una interfaz de servicios web que guarda y recupera la cantidad de información que el usuario almacene y desde cualquier ubicación del planeta. Ofrece una disponibilidad de 99.9999999% y es masivamente escalable. Usando el API tipo Rest de Amazon Web Service y su autenticación basada en *Hash Message Authentication Code* (keyed-HMAC) asegura que la subida de datos sea desde una fuente confirmada. (Amazon, 2017)

### **3.Capítulo III. Desarrollo de la Aplicación**

#### **3.1 Situación Actual**

Este capítulo define cuál es la serie de pasos para dos trabajadores artesanos que actualmente ofrecen sus productos por internet a través de dos medios populares, Facebook y Wordpress.

Para Kuma, una productora ecuatoriana de mermelada artesanal que usa a Facebook para promocionar sus productos, el proceso comienza creando o seleccionando una publicación para ser promovida, a continuación selecciona el alcance que tendrá la publicación y procederá a ingresar su información para el pago, cuando el pago ha sido realizado el anuncio estará visible como propaganda.

Las personas que se encuentran interesadas escriben a Kuma usando la mensajería de Facebook para que sus inquietudes sean atendidas.

Además del tiempo para promocionar una publicación se añade el tiempo que tomó crear una página de su empresa en Facebook de la cual generar publicaciones.

Para Dunamis, una productora ecuatoriana de joyería artesanal que usa el CMS WordPress para promocionar sus productos, el proceso comienza con añadir una nueva sección en su página mediante el manejador de contenido, subir una serie de imágenes referentes a los productos y agregar una descripción, una vez actualizado su sitio, publica en su cuenta de Facebook la página para que alcance más audiencia.

La página de WordPress de Dunamis tiene un formulario para que los usuarios se contacten por medio de correo electrónico, que no muestra gran acogida ya que sólo ha recibido un máximo de tres correos electrónicos por mes.

En la Tabla 2 se muestra el resultado en ventas, en número de personas que alcanzó la publicación y los recursos invertidos en tiempo y dinero, que serán índices para la situación actual de los artesanos mencionados.

Tabla 2

*Índices de los procesos de promoción de productos artesanales mediante los medios de Facebook y WordPress*

	<b>Kuma - Facebook</b>	<b>Dunamis - Wordpress</b>
<b>Tiempo Invertido</b>	1 hora	3 horas
<b>Gastos</b>	\$15	\$12
<b>Ventas</b>	60 frascos	20 piezas
	<b>Kuma - Facebook</b>	<b>Dunamis - Wordpress</b>
<b>Número aproximado de personas alcanzadas</b>	1300 personas	700 personas

### 3.2 Aplicación Propuesta

La aplicación propuesta en el alcance de la Introducción de este documento, se desarrollará usando SCRUM, para este proceso se ha definido los roles en la Tabla 3.

Tabla 3

*Responsables proceso SCRUM*

<b>Rol</b>	<b>Responsable</b>
Scrum Master	Santiago Villarreal
Product Owner	Alberto Tatés
Scrum Team	Alberto Tatés

<b>Rol</b>	<b>Responsable</b>
Cliente	Dunamis

Se asigna un tiempo para cada Sprint de una semana laborable y se trabajará con una velocidad de cinco puntos de historia por Sprint, esta velocidad se ha tomado en cuenta por el SRUM Team mediante estimación. El respectivo seguimiento del proyecto se realiza en el siguiente enlace de la herramienta PivotalTracker: <https://www.pivotaltracker.com/n/projects/2001597>.

### **3.1 Planificación o Sprint 0**

A partir de los requerimientos obtenidos y del alcance propuesto del proyecto, se suman los resultados de la situación actual y se recopila con el equipo SCRUM la siguiente lista de requerimientos:

1. La aplicación debe tener una entrada de texto para ingresar una palabra por la cual se realizará una búsqueda entre los productos ingresados.
2. En el resultado de la búsqueda la aplicación permite navegar a una nueva página con la información completa referente del producto en la que se encuentra el usuario que lo subió con una lista para filtrar los resultados obtenidos.
3. La aplicación permite navegar a una nueva página donde está la información del usuario y una opción para ver si el usuario está conectado.
4. La aplicación permite iniciar una sesión de chat con el usuario.
5. La aplicación permite ver la bandeja de mensajes que han sido enviados y recibidos.
6. La aplicación permite crear usuarios con Facebook.
7. La aplicación permite crear usuarios con email y contraseña.
8. La aplicación permite que un usuario registre su producto o servicio.

Tomando en cuenta las dependencias entre los requerimientos se lista en la Tabla 4 siete historias de usuario seguidas por la prioridad o la cantidad de valor que agrega a los usuarios, marcadas por el cliente junto con el *Product*

*Owner* donde uno es el más importante y cuatro el menos importante, también el puntaje de historia o la complejidad que lleva desarrollar cada una, siguiendo una escala Fibonacci donde uno es el más bajo y ocho el más alto.

Tabla 4

*Historias de usuario de acuerdo a su valor y a los puntos asignados*

<b>Historia de Usuario</b>	<b>Prioridad</b>	<b>Punto de historia</b>
El usuario puede realizar una búsqueda	2	5
El usuario puede ver la información de un producto	4	3
El usuario puede ver la información de un artesano y si este se encuentra en línea	2	3
El usuario puede iniciar una sesión de chat con un artesano	1	8
El usuario puede ver su bandeja de mensajes	3	8
El usuario puede registrarse con su email y contraseña y usando Facebook	2	8
El usuario puede registrar un producto	2	8
Total		43
Tiempo aproximado		9 semanas

A partir de la estimación para cada historia de usuario se genera el Product Backlog mostrado en la Tabla 5, con preferencia a las historias con mayor prioridad y menor complejidad, se toma en cuenta también la dependencia entre historias y se asigna el Sprint correspondiente. La Figura 5 muestra el Product Backlog ingresado en la herramienta PivotalTracker.

Tabla 5

*Product Backlog*

<b>Id</b>	<b>Sprint</b>	<b>Puntos de historia</b>	<b>Historia</b>
1	1 - 2	8	El usuario puede registrarse con su email y contraseña y usando Facebook
2	2 - 3	3	El usuario puede ver la información de un artesano y si este se encuentra en línea
3	3 - 4	8	El usuario puede iniciar una sesión de chat con un artesano
4	4 - 5 - 6	8	El usuario puede ver su bandeja de mensajes
5	6 - 7	8	El usuario puede registrar un producto o servicio
6	8	5	El usuario puede realizar una búsqueda
7	9	3	El usuario puede ver la información de un producto



Figura 5. Backlog de las historias de usuario en la herramienta PivotalTracker

### 3.2 Historia de Usuario uno

Nombre: El usuario puede registrarse con su email y contraseña o usando Facebook.

Descripción: Desde la página principal de la aplicación, se puede dirigir a una página de registro donde ingresando un correo y una contraseña se puede registrar, también se puede usar Facebook, de donde se obtendrá información básica del usuario. Una vez registrado el usuario puede ingresar más información.

Tareas a realizar:

1. Investigar acerca de un paquete de Meteor que permita hacer registro.
2. Agregar *Iron-Router* para manejar las rutas de la página.
3. Personalizar la forma de registro y la forma de *login*.

4. Crear dos usuarios nuevos, salir de la aplicación y volver a ingresar.
5. Agregar las secciones para ingresar información de usuario y de artesano.
6. Agregar AWS para almacenamiento para las imágenes.
7. Agregar Google *Recaptcha* para verificar que no es un robot.
8. Agregar *Materialize* para estilizar la página.
9. Agregar un elemento para verificar el mail.
10. Desplegar la aplicación en Heroku.

Criterios de aceptación:

1. La aplicación tiene una nueva página de registro.
2. La aplicación permite ingresar el correo y la contraseña para crear el usuario.
3. La aplicación tiene los elementos para ingresar el email y la contraseña y muestra si el usuario está ingresado.
4. La aplicación muestra un icono de Facebook para poder ingresar.
5. En la cuenta del usuario se ve si el mail ha sido verificado o no y hay un control que envía un correo para verificarlo.
6. La aplicación tiene dos campos para subir fotos, una para el usuario y otra para el artesano.
7. En la cuenta del usuario están elementos para ingresar información de : nombre de usuario, apellido de usuario, número telefónico, nombre de artesano, dirección de artesano, ciudad de artesano con Google Maps, servicio a domicilio, foto, contacto de artesano: teléfono, correo electrónico, pagina web, foto.

Esta historia tomó más de los dos Sprints, debido a la dificultad en la implementación de la tarea seis. En la Tabla 6 se listan los Sprints usados y las tareas realizadas para cada uno.

Tabla 6

*Desarrollo de las tareas de la historia uno en cada Sprint utilizado*

Sprint	Tarea
1	1, 2, 3, 4
2	4, 5, 6
3	7, 8, 9, 10

Al finalizar esta historia, se cumplen los criterios de aceptación y un usuario ya puede registrarse mediante la forma que se muestra en la Figura 6. También se puede editar la información del usuario desde el formulario que se ve en la Figura 7.

**Bienvenid@**

**Regístrate para entrar al mundo Craft!**

Puedes usar Facebook para hacerlo mas rápido

**Registro**

Email

Contraseña

Repita contraseña

Acepto [los términos y condiciones](#)

No soy un robot 

INGRESAR [USAR FACEBOOK](#)

**Ingresar**

Email

Contraseña

[ENTRAR](#) o usa 

[Olvide mi contraseña?](#)

*Figura 6. Formulario de registro, entrega de la historia de usuario uno*

**Información de Usuario**

CAMBIAR FOTO

Nombre

Apellidos

Teléfono

asf@asd.com **Tu correo electrónico no está verificado**  
click aquí para confirmar

ACTUALIZAR

*Figura 7.* Formulario de información de usuario, entrega de historia de usuario uno

### 3.3 Historia de usuario dos

**Nombre:** El usuario puede ver la información de un artesano y si este se encuentra en línea.

**Descripción:** Al acceder a la página de un artesano, se puede ver su información y se encuentra en línea.

**Tareas a realizar:**

1. Crear la ruta para el perfil de cada usuario.
2. Agregar la lista de los usuarios en la página principal.
3. Hacer que se vea si el usuario está conectado o no.
4. Restringir las rutas para cuando un usuario no está conectado.

**Criterios de aceptación:**

1. Se puede acceder al perfil de un usuario.

2. Se puede ver si se encuentra conectado o no.
3. Se puede ver toda la información de un usuario dependiendo si es artesano o no.
4. Se puede ver la lista de sus productos o servicios.

Esta historia tomó parte del Sprint número cuatro para ser realizada. En la Figura 8 se puede observar la página de un artesano y al lado de su nombre un indicador que muestra si está actualmente conectado.

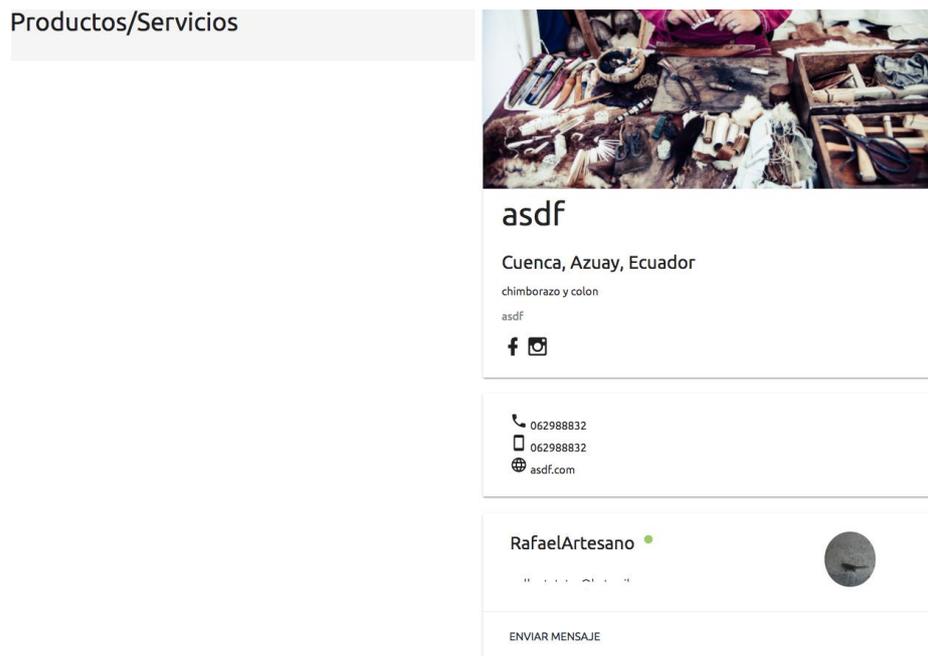


Figura 8. Página de perfil de un artesano, entrega de historia de usuario dos

### 3.4 Historia de usuario tres

Nombre: El usuario puede iniciar una sesión de *chat* con un artesano.

Descripción: El usuario puede escribir un mensaje a un artesano y ver como en la página del *chat* se muestran los nuevos mensajes.

Tareas a realizar:

1. Agregar las colecciones de *room* y de mensajes para la base de datos.
2. Crear las funciones para permitir verificar un *room* y agregar un mensaje nuevo.

3. Crear la vista para poder ver y enviar los mensajes.
4. Agregar un elemento en la página principal que permite ver las notificaciones.
5. Hacer que el elemento que muestra que un usuario está conectado abra la ventana de mensajes.
6. Configurar la ruta para marcar un mensaje como visto.

Criterios de aceptación:

1. A partir de la página de perfil de un artesano, un usuario puede dar *click* al icono que indica que el artesano está en línea para dirigirse a una ventana de *chat* donde se pueden ver los mensajes.
2. El usuario puede ver mensajes y ver como los mensajes se actualizan.
3. En la página principal debe haber un elemento donde se vean las notificaciones de nuevos mensajes.
4. Cuando se recibe se debe notificar al usuario.

Esta historia tomó el resto del Sprint cuatro y también el Sprint cinco para ser desarrollada, el mayor esfuerzo empeñado fue en la tarea dos que conlleva la lógica de manejar un servicio de mensajería. En la Tabla 7 se listan los Sprints usados y las tareas realizadas para cada uno.

Tabla 7

*Desarrollo de las tareas de la historia de usuario tres en cada Sprint utilizado*

Sprint	Tarea
4	1, 2
5	3, 4, 5, 6

Al finalizar esta historia, se cumplen los criterios de aceptación y un usuario puede escribir y recibir mensajes como muestra la Figura 9.



Figura 9. Elemento de mensajería instantánea, entrega historia de usuario tres

### 3.5 Historia de usuario cuatro

Nombre: El usuario puede ver su bandeja de mensajes

Descripción: El usuario tiene una lista de las conversaciones que ha hecho y al seleccionar una de las listas se muestran todos los mensajes correspondientes.

Tareas a realizar:

1. Crear las rutas para dirigirse a la página de mensajes.
2. Crear la página para ver los mensajes agregando una sección para la lista de mensajes y otra para los mensajes en sí.
3. Actualizar dinámicamente cuando se seleccione una conversación para ver los mensajes.
4. Hacer que el elemento con los mensajes se haga un *scroll* hacia los más actuales.
5. Permite ver la fecha y hora cuando los mensajes fueron enviados.

Criterios de aceptación:

1. Existe una página dedicada únicamente para los mensajes.
2. La página muestra todas las conversaciones existentes.

3. Al dar *click* en una conversación la página muestra los mensajes adecuados.
4. La lista de mensajes muestra si hay una conversación nueva.

Esta historia tomó parte del Sprint número cinco para ser realizada. En la Figura 10 se puede observar las conversaciones que actualmente tiene un usuario.



*Figura 10.* Bandeja de mensajes de un usuario, entrega de la historia de usuario cuatro

### 3.6 Historia de usuario cinco

Nombre: El usuario puede registrar un producto o servicio.

Descripción: En la página de la cuenta del usuario, la aplicación le permite ingresar un nuevo producto o servicio, ver los productos que tiene, editarlos y eliminarlos.

Tareas a realizar:

1. Crear la vista para ingresar un producto.
2. Crear la paginación para las partes de la creación del producto.
3. Crear formularios para la creación del producto.
4. Crear la vista para el resumen de publicaciones.
5. Permitir seleccionar una publicación para editarla.
6. Crear las vistas para editar las publicaciones.
7. Crear un botón para eliminar la publicación.

Criterios de aceptación:

1. La página presenta un espacio donde el usuario elige si va a publicar un servicio o un producto.
2. Según el tipo de publicación en la página muestra el primer formulario para llenar la información general.
3. Un segundo formulario para la información extra.
4. Un tercer formulario para las imágenes.
5. Una página que permite ver al usuario todas las publicaciones.
6. Las páginas que permiten al usuario editar una publicación en una nueva vista.
7. La página permite eliminar una publicación seleccionada.

Esta historia tomó el resto del Sprint cinco y también el Sprint seis para ser desarrollada. En la Tabla 8 se listan los Sprints usados y las tareas realizadas para cada uno.

Tabla 8

*Desarrollo de las tareas de la historia de usuario cinco en cada Sprint utilizado*

Sprint	Tarea
5	1, 2, 3
6	4, 5, 6, 7

Al finalizar esta historia, se cumplen los criterios de aceptación y un usuario puede crear un producto o servicio comenzando con la forma de información básica que se muestra en la Figura 11 y también listando todas las publicaciones con las respectivas acciones que se pueden tomar, como se ve en la Figura 12.

**Información general**

Etiqueta/as  
servicio x X

Nombre  
servicio x

Descripción  
el servicio esta bueno

SIGUIENTE ▶

Figura 11. Formulario de información general de una publicación, entrega de la historia de usuario cinco

#### Servicios



**servicio x**

servicio x

el servicio esta bueno

Pago por contrato

EDITAR VISTA PREVIA ELIMINAR COMPARTIR

DEJAR DE PUBLICAR

#### Productos



Figura 12. Vista de las publicaciones de un usuario, entrega de la historia de usuario cinco

### 3.7 Historia de usuario seis

Nombre: El usuario puede realizar una búsqueda.

Descripción: El usuario puede escribir una palabra clave desde la página principal y la página se dirige a la vista de búsquedas donde se muestran los resultados a partir de la palabra ingresada además de filtros que pueden editar los resultados de la búsqueda.

Tareas a realizar:

1. Agregar la publicación del servidor para obtener resultados de publicaciones a partir de la palabra ingresada.
2. Agregar la ruta para la página que presenta los resultados
3. Agregar la barra de búsqueda y configurar para que realice la redirección
4. Agregar elementos en la página de resultados para poder filtrar la información
5. Hacer que los parámetros se guarden en el URL.

Criterios de aceptación:

1. Existe un buscador en la página principal que permite dirigirse a las búsquedas.
2. La página de búsquedas muestra los resultados a partir de la palabra escrita.
3. Existe una barra de búsqueda en la barra de navegación principal.
4. La página de los resultados permite filtrar los resultados por: tipo, ubicación, etiquetas, es negociable, tiene promoción, el envío es incluido.
5. La página de resultados tiene otra barra de búsquedas que permite ingresar más palabras.

Esta historia tomó el Sprint siete y parte del Sprint 8 para ser desarrollada, la Tabla 9 muestra las tareas de la historia y en que Sprint fueron implementadas.

Tabla 9

Desarrollo de las tareas de la historia de usuario seis en cada Sprint utilizado

Sprint	Tarea
7	1, 2
8	3, 4, 5

Al finalizar esta historia se cumplen los criterios de aceptación y un usuario puede ingresar una palabra clave la cual refleja un resultado de acuerdo a la descripción de la publicación, a su ubicación, el nombre y las etiquetas que esta tenga. En la Figura 13 se observa el resultado de la palabra “cuenca”.

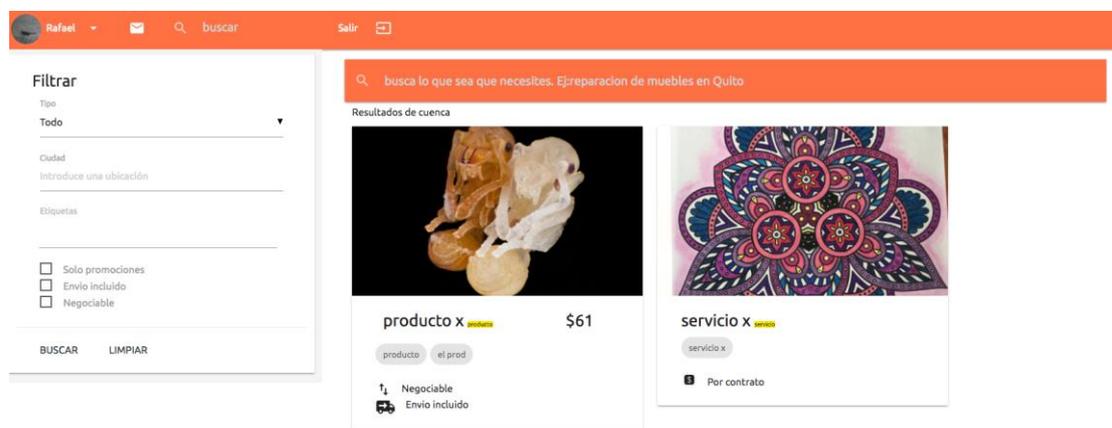


Figura 13. Resultados de búsqueda con la palabra “cuenca”, entrega de la historia de usuario seis

#### 4.6 Historia de usuario siete

Nombre: El usuario puede ver la información de un producto o servicio.

Descripción: El usuario puede dar *click* en un producto de la búsqueda y ver su información completa junto con la de su vendedor.

#### Tareas a realizar:

1. Crear la ruta para la vista de información del producto o servicio.
2. Agregar toda la información correspondiente diferenciando al producto del servicio.
3. Agregar la información del usuario que publica.
4. Agregar un botón para ver la información completa del usuario que publica y todas sus publicaciones.

#### Criterios de aceptación:

1. Al dirigirse a una ruta exclusiva de la publicación, este muestra toda la información y se diferencia si es un servicio o producto.
2. La información también muestra al perfil que está haciendo la publicación.
3. La página permite dirigirse al perfil del usuario para ver la información completa.

Esta historia tomó el Sprint ocho para ser desarrollada, al finalizar esta historia se cumplen los criterios de aceptación y un usuario puede ver la información de un producto y servicio, donde se muestran las fotos del producto en tipo diapositiva como se ve en la Figura 14.

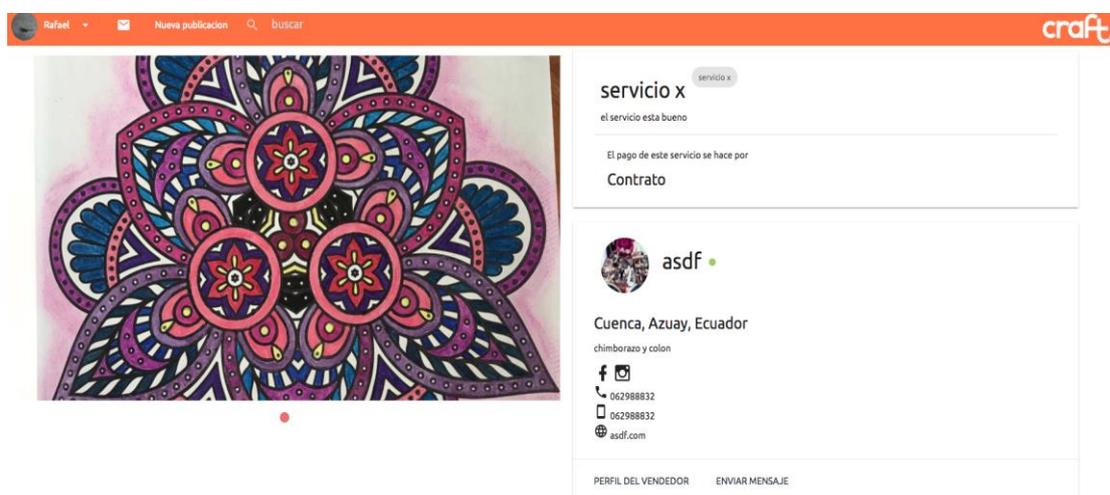


Figura 14. Página de un producto o servicio, entrega de la historia de usuario ocho

#### 4. Capítulo IV. Resultados

El resultado final es una aplicación a la que se llamó Craft.ec, desplegada para el Internet en el siguiente enlace [www.craft-ec.com](http://www.craft-ec.com) que permite a personas naturales buscar por productos y servicios artesanales y a los artesanos publicar su trabajo. El manual de usuario de la aplicación se encuentra en el Anexo A de este documento.

La aplicación se presentó a los casos de la Situación Actual, para que realicen la misma operación de publicar un producto. Los resultados obtenidos se muestran en la Tabla 10, debido al poco tiempo de la aplicación y las publicaciones en el aire los resultados en cuanto a la cantidad de ventas no se muestran en la tabla ya que no fueron medibles.

Tabla 10

*Resultados*

	Kuma		Diego	
	Craft.ec	Facebook	Craft.ec	Wordpress
<b>Tiempo Invertido</b>	20 minutos	1 hora	14 minutos	3 horas
<b>Gastos</b>	\$0	\$15	\$0	\$12

Ambos casos reconocieron la facilidad de usar la mensajería instantánea para recibir mensajes de otros usuarios.

Técnicamente, la aplicación responde a criterios de:

- Usabilidad. Al usar un *framework* para la interfaz de usuario este presenta la información de una manera optimizada e intuitiva.
- Seguridad. Al usar un proveedor tercero para la autenticación y usar una base de datos con un esquema no relacional.
- Disponibilidad. Al usar servidores en la nube tanto para el host como para la base de datos y el almacenamiento de multimedia que brindan una disponibilidad de 99%.

- Integridad. Meteor permite optimizar las funciones para realizar actualizaciones en la base de datos, negando el paso hacia un documento que no le pertenezca al usuario correspondiente.

Para generar recursos económicos que sirvan para la aplicación, se espera posteriormente, agregar un módulo donde los artesanos puedan contratar un servicio *Premium* para que sus publicaciones aparecen como destacadas en la aplicación.

## 5. Conclusiones y Recomendaciones

### 5.1 Conclusiones

Con el *framework* Meteor se reduce el tiempo de configuración de la persistencia entre los datos del *front-end* y los de la base de datos, ya que la base de datos se ve reflejada en el cliente una *query* de búsqueda o actualización puede ser ejecutada desde el buscador.

Los paquetes disponibles para aplicaciones de Meteor agilizan mucho el desarrollo, ya que múltiples características (*features*) como el registro de usuarios se encuentran disponibles para no tener que construir desde cero.

Desplegar una aplicación en Heroku tiene una dificultad baja ya que sólo necesita un paquete de construcción que puede ser encontrado para los diferentes *frameworks* del mercado, desde la consola de Heroku se asigna un repositorio para que el despliegue se haga automático una vez que el repositorio haya cambiado.

Meteor renderiza (proceso de generar imágenes a partir de lenguaje computacional) las vistas del lado del cliente, esto tiene la ventaja de velocidad y de liberación de carga para el servidor. Es importante para que la indexación de buscadores funcione correctamente especificar con un *Meta Tag* que se pida por una renderización en el servidor.

Las funciones de *publish* y *subscribe* de Meteor son esenciales ya que se encargan de reflejar la base de datos desde el servidor hacia el cliente, entonces estas dos funciones se convierten en controladores para resguardar qué datos son los que se ponen a disposición del cliente así agregan seguridad basada en la lógica del negocio.

Meteor permite crear índices en la base de datos desde los archivos del servidor, cuando la aplicación se ejecuta verificará o creará el índice de acuerdo a las especificaciones. Es importante cerciorarse si MongoDB no permite más de dos índices de la misma clase en una colección para no tener errores de sobre escritura si se modifica la configuración del índice.

La metodología ágil SCRUM se acopla de manera muy adecuada a este tipo de proyectos orientado a generar valor con cada Sprint, conforme se planifica y desarrolla un Sprint van surgiendo nuevas necesidades tanto técnicas como para la lógica del negocio que a pesar de extender el proyecto generan un resultado final mejorado.

## **5.2 Recomendaciones**

En cuanto a multimedia, Heroku como servidor limita la capacidad de almacenamiento por lo que se recomienda optar por otro servicio. Entre Amazon Web Services y Google Cloud Storage es recomendado usar la primera opción por la comodidad en cuanto a costos y los múltiples paquetes existentes para interactuar con su API.

Se recomienda utilizar Meteor como framework, no únicamente si se tiene una necesidad de tener un flujo de datos en tiempo real, sino para cualquier tipo de aplicación web o móvil. Meteor ofrece mucha facilidad para construir aplicaciones sin tener que concentrarse muy minuciosamente en tareas generales como la configuración conexiones a bases de datos.

Se recomienda usar un índice de texto si se piensa realizar una búsqueda sobre texto en los documentos de alguna colección de MongoDB. Esto hace que la búsqueda de texto sea muy rápida y precisa, también permite ordenar los datos de acuerdo a los documentos que más encajaron con el criterio a buscar.

Para aplicar estrategias de SEO se recomienda usar el paquete de Meteor *spiderable* el cual permite que las vistas sean renderizadas en el servidor, este paquete usa PhantomJS para realizar estas operaciones por lo que se recomienda mantener siempre versiones actuales de estos productos.

Se recomienda usar conjuntamente GitHub como manejador de versiones y Heroku como *host* de la aplicación para poder relacionarlos y configurar a Heroku con despliegue continuo dependiendo de los *commits* del repositorio.

A la hora de manejar imágenes que se suben al servidor, se recomienda usar nombres diferentes para cada URL de descarga, de lo contrario la imagen quedará guardada en la memoria caché del buscador y no se presentará la imagen a la que el enlace actualmente apunta.

## Referencias

Amazon. (s,f). Amazon S3. Recuperado el 6 de Julio de 2017 de <https://aws.amazon.com/es/s3/>

Developer Apple. (2016). *Model View Controller*. Recuperado el 10 de Diciembre de 2016 de <https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>

Heroku. (s,f). Heroku. Recuperado el 6 de Julio de 2017 de <https://www.heroku.com/>

IESS. (2017). IESS, MRL, SRI y JNDA en inspecciones para el sector artesanal. Recuperado el 30 de Enero de 2017 de [https://www.iess.gob.ec/en/sala-de-prensa/-/asset\\_publisher/4DHq/content/iess-mrl-sri-y-jnda-en-inspecciones-para-el-sector-artesanal/10174?redirect=https%3A%2F%2Fwww.iess.gob.ec%2Fen%2Fsala-de-prensa%3Fp\\_p\\_id%3D101\\_INSTANCE\\_4DHq%26p\\_p\\_lifecycle%3D0%26p\\_p\\_state%3Dnormal%26p\\_p\\_mode%3Dview%26p\\_p\\_col\\_id%3Dcolumn-1%26p\\_p\\_col\\_pos%3D1%26p\\_p\\_col\\_count%3D2%26\\_101\\_INSTANCE\\_4DHq\\_advancedSearch%3Dfalse%26\\_101\\_INSTANCE\\_4DHq\\_keywords%3D%26\\_101\\_INSTANCE\\_4DHq\\_delta%3D6%26\\_101\\_INSTANCE\\_4DHq\\_cur%3D71%26\\_101\\_INSTANCE\\_4DHq\\_andOperator%3Dtrue?mostrarNoticia=1](https://www.iess.gob.ec/en/sala-de-prensa/-/asset_publisher/4DHq/content/iess-mrl-sri-y-jnda-en-inspecciones-para-el-sector-artesanal/10174?redirect=https%3A%2F%2Fwww.iess.gob.ec%2Fen%2Fsala-de-prensa%3Fp_p_id%3D101_INSTANCE_4DHq%26p_p_lifecycle%3D0%26p_p_state%3Dnormal%26p_p_mode%3Dview%26p_p_col_id%3Dcolumn-1%26p_p_col_pos%3D1%26p_p_col_count%3D2%26_101_INSTANCE_4DHq_advancedSearch%3Dfalse%26_101_INSTANCE_4DHq_keywords%3D%26_101_INSTANCE_4DHq_delta%3D6%26_101_INSTANCE_4DHq_cur%3D71%26_101_INSTANCE_4DHq_andOperator%3Dtrue?mostrarNoticia=1)

Materialize. (s,f). *Materialize*. Recuperado el 6 de Julio de 2017 de <http://materializecss.com/>

Ramos, A. y Ramos. (2013). *M. Aplicaciones Web*. (2.<sup>a</sup> ed.) Madrid, España: Ediciones Paraninfo.

Rouse, M. (s.f). *What is a content management system*. Recuperado el 8 de Diciembre de 2016 de <http://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS>

Owens, J. (2015). *WHAT IS METEOR.JS?*. Recuperado de <http://jshowens.me/what-is-meteor-js/> el 12 de Diciembre de 2016

Manchón, E. (2013). Desarrollo ágil. Recuperado el 20 de Diciembre de 2016 de <https://www.desarrolloweb.com/articulos/desarrollo-agil.html>

Meteor. (2017). *Community*. Recuperado de <https://www.meteor.com/> el 30 de Enero de 2017

Proyectos Ágiles. (s.f). Qué es SCRUM. Recuperado de <https://proyectosagiles.org/que-es-scrum/> el 20 de Marzo de 2017

PivotalTracker. (s.f). *Product* .Recuperado el 20 de Marzo de 2017 de <https://www.pivotaltracker.com/product>

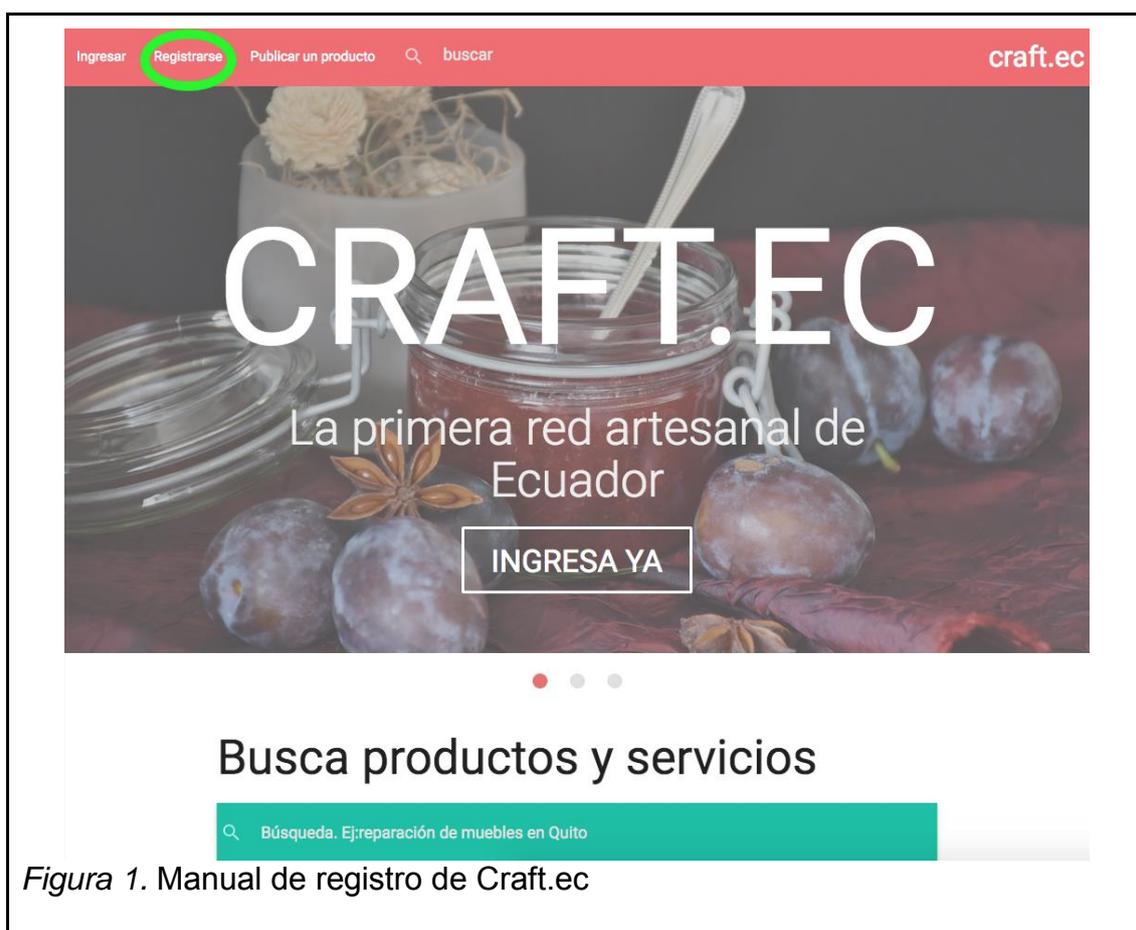
## **ANEXOS**

## A. Manual de usuario

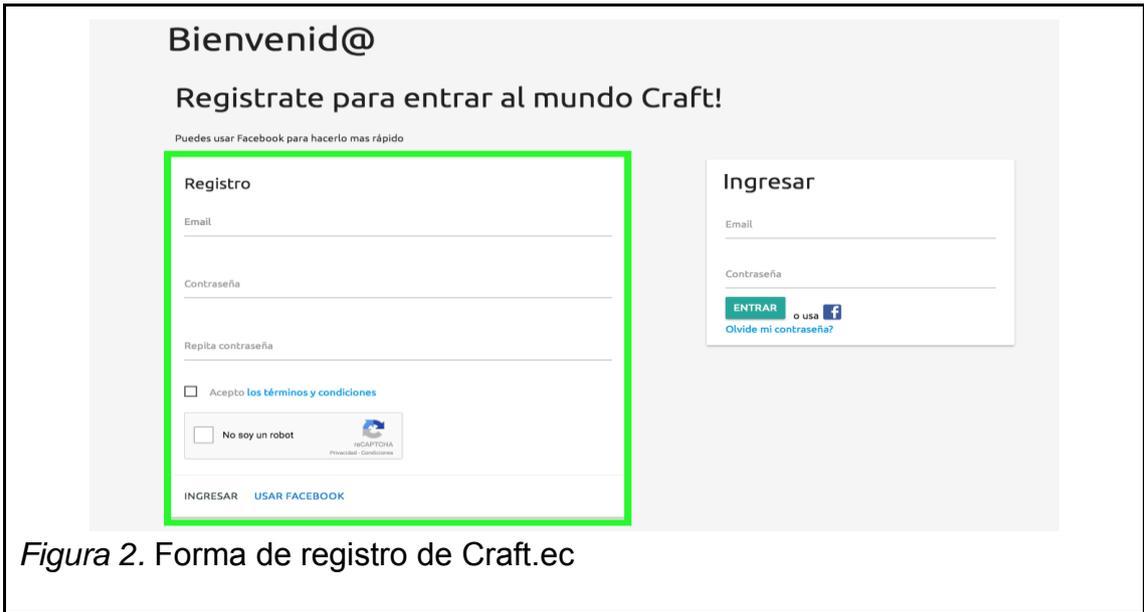
El presente manual describe el correcto uso de la aplicación.

### 1. Creación de usuario

En la pantalla principal haciendo *click* en el botón de la barra superior donde dice “Registrarse” como muestra la Figura 1.



Este dirigirá a la página de registro, donde se muestra el formulario de registro como se ve en la Figura 2. Al llenar la forma, marcar que se acepta los términos y condiciones y que no es un robot dando *click* en “Ingresar” se creará un nuevo usuario.



Craft.ec allows you to register using your electronic mail and a password or your Facebook account.

### 1.1 Register with Facebook

It is also possible to register with the data of a Facebook account, for this you can click on any of the buttons marked in Figure 3.



A continuación se desplegará una pantalla de confirmación por parte de Facebook como se muestra en la Figura 4.

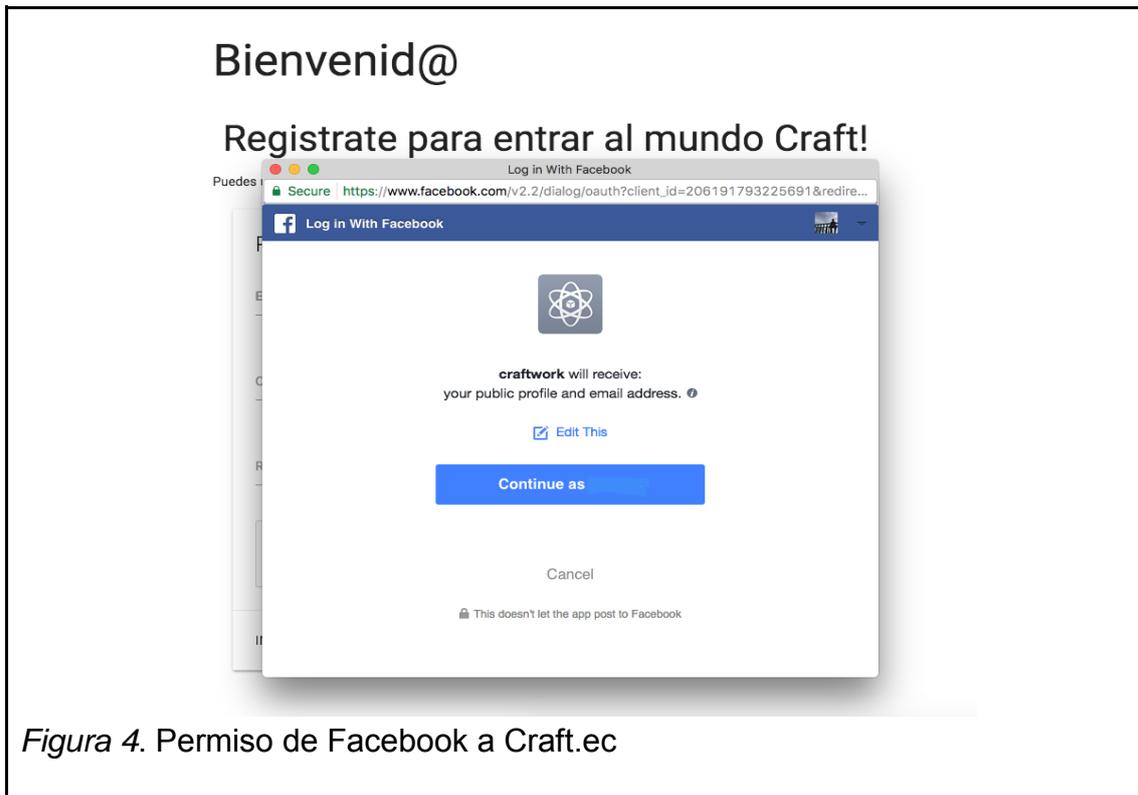


Figura 4. Permiso de Facebook a Craft.ec

Una vez confirmado, Facebook direccionará hacia Craft.ec con el nuevo usuario ya ingresado.

## 2. Verificar dirección correo electrónico

Si un usuario se ha registrado con correo y una contraseña es indispensable que verifique la dirección de correo electrónico para usar la plataforma en tareas como

- Enviar mensajes.
- Publicar productos o servicios.

Cuando un usuario está *logueado* en la aplicación, desde la página principal haciendo *click* al botón que tiene el nombre del usuario como muestra la Figura 5, y seleccionando la opción que dice cuenta se dirigirá a la página de información del usuario.



Figura 5. Mi cuenta desde la página principal Craft.ec

En esta sección de la información personal en la parte que dice que “tu correo no está confirmado”, al hacer “click” en “click aquí para confirmar” como muestra la Figura 6 un correo de confirmación se enviará a esa dirección.

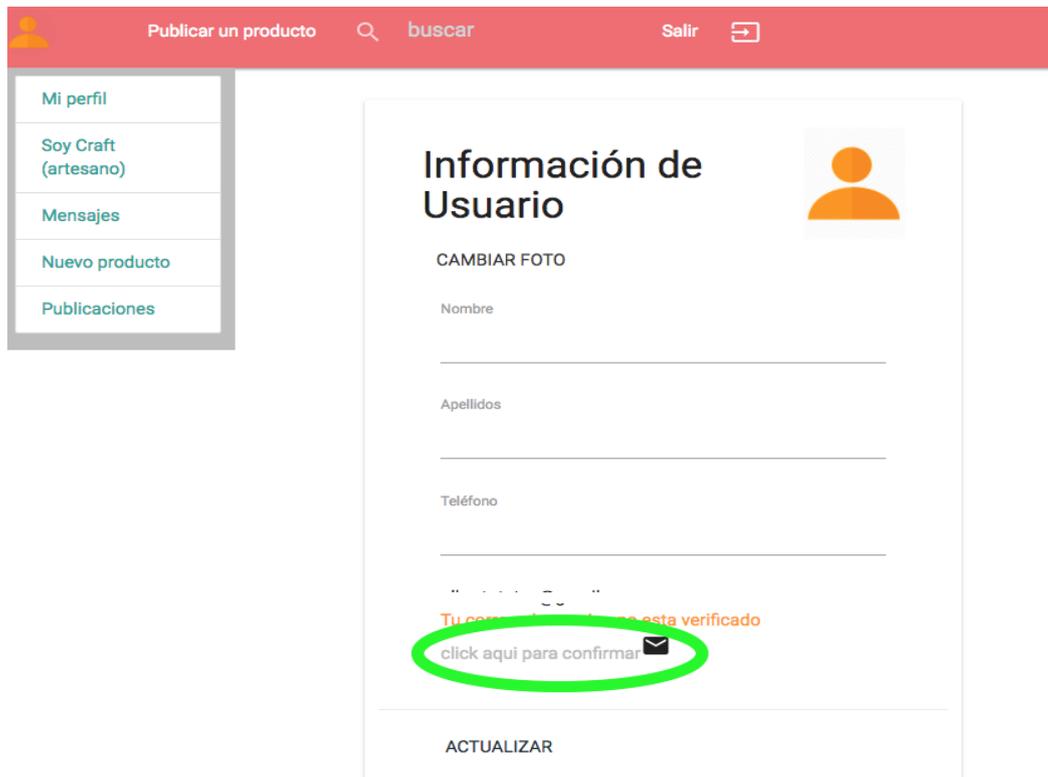
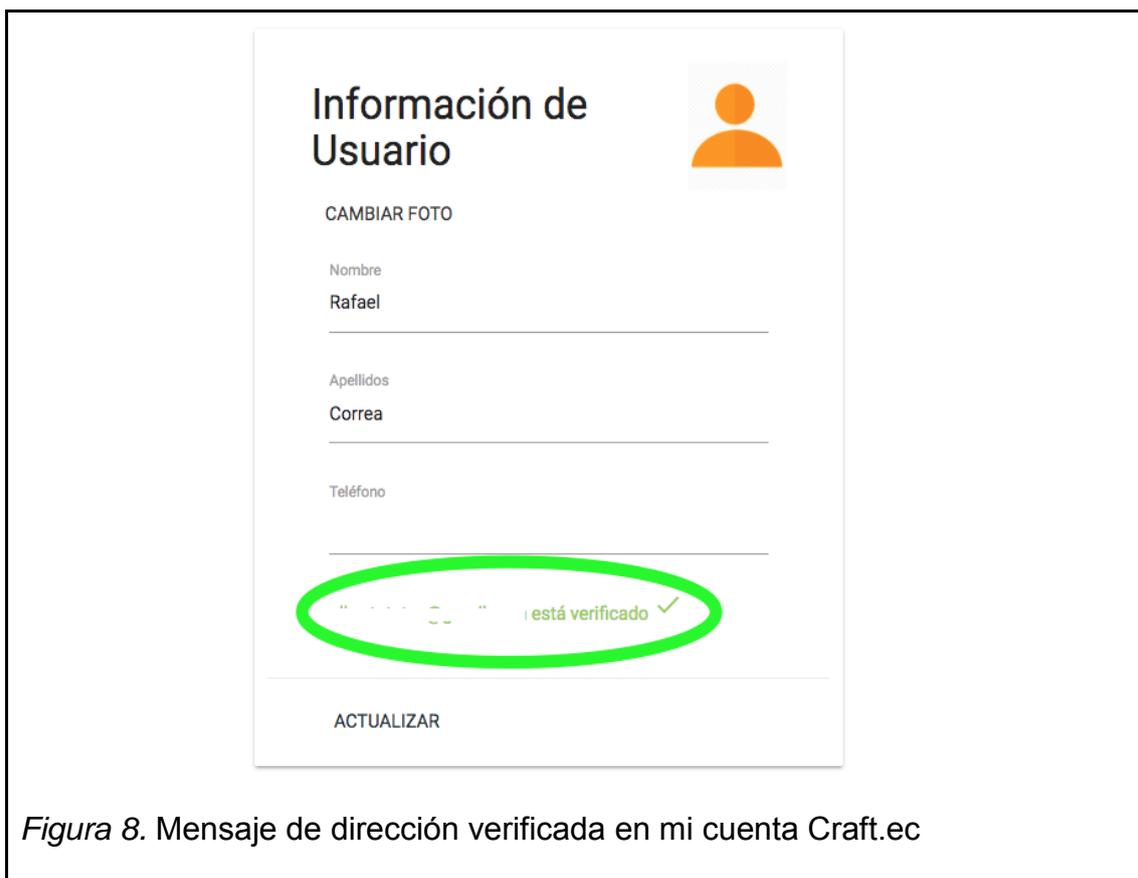


Figura 6. Confirmar correo desde mi cuenta Craft.ec

En el proveedor de correo electrónico del usuario debe haber un mensaje desde la dirección [craftec.customerservice@gmail.com](mailto:craftec.customerservice@gmail.com), que en su contenido tendrá un enlace que el usuario deberá dar “click” para ser redirigido como muestra la Figura 7.



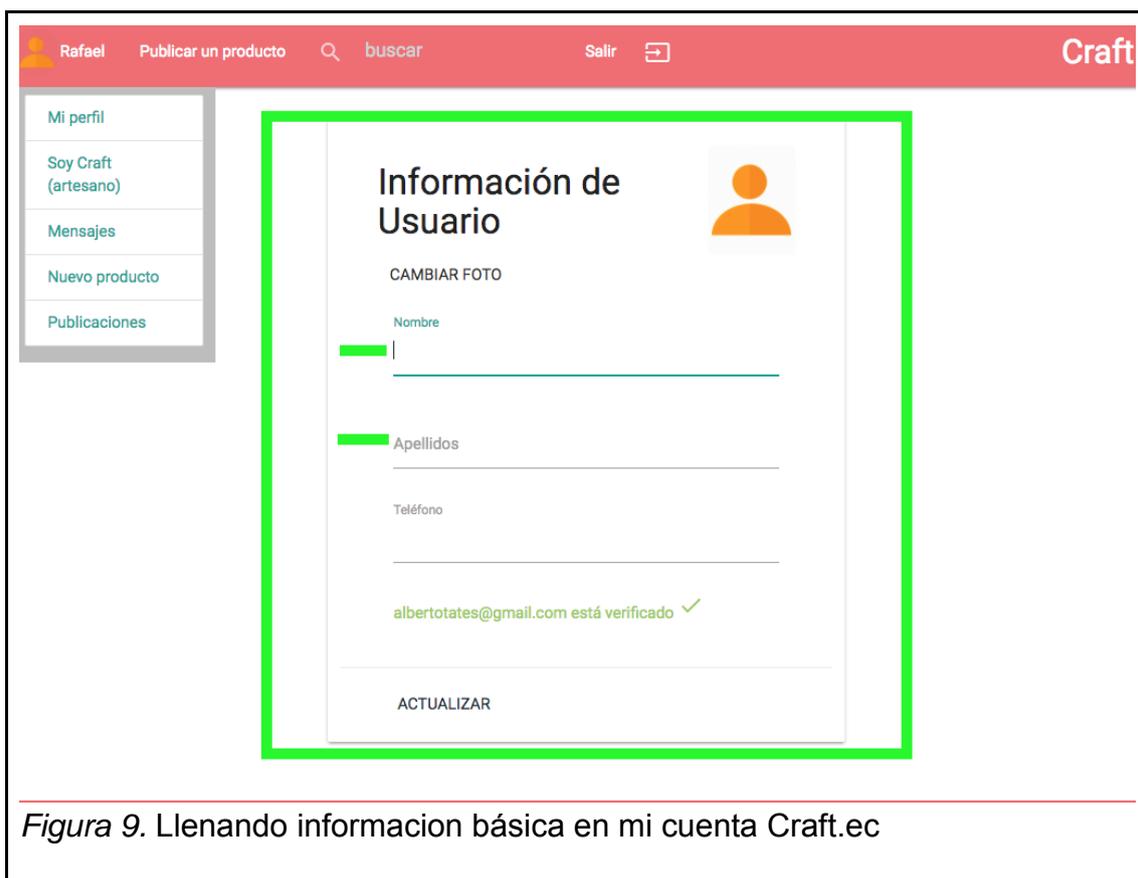
La redirección llevará de nuevo a Craft.ec donde un mensaje se mostrará, y si el usuario se dirige a su información, podrá ver su dirección de correo marcada como confirmada como muestra la Figura 8.



### 3. Crear un producto o servicio

Para crear un producto o servicio es indispensable que el usuario haya llenado la información personal y la información de artesano, para ello desde la pantalla principal, se debe hacer *click* en el botón donde está el nombre del usuario, luego en la opción de “mi cuenta”.

En esta sección como se muestra en la Figura 9 se debe rellenar obligatoriamente el nombre y apellido, si el usuario desea, puede cambiar su foto de perfil dando “click” al botón que dice “Cambiar Foto”.



Dando *click* en “Actualizar” la información quedará guardada.

A continuación se debe llenar la información de artesano, para esto se debe dar *click* en el botón que dice “Soy Craft (artesano)” como se muestra en la Figura 10 en el menú de la izquierda.



Figura 10. Sección de artesano de Craft.ec

En la sección de artesano, se debe rellenar obligatoriamente los campos de:

- Nombre artesanal / empresa
- Descripción
- Ciudad, escogiendo una en la lista que se despliega
- Dirección de donde se encuentra
- Un teléfono de contacto

Se puede cambiar la foto de presentación de artesano o dejar la que está por defecto.

Cuando la información esté completa dando *click* en el botón de *Actualizar* como muestra la Figura 11 se guardará la información.

## Información de Artesano

Nombre artesanal / empresa

Describelo  
Qué ofreces?

Ciudad

Dirección

Teléfono

Teléfono secundario

Perfil en web

Facebook

Instagram

Foto

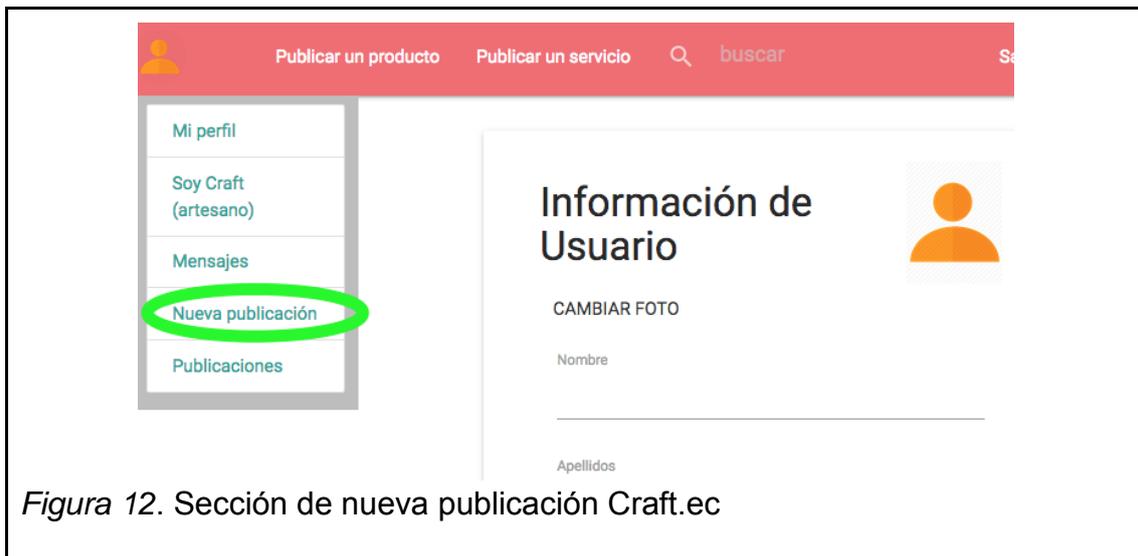


CAMBIAR

ACTUALIZAR

Figura 11. Formulario de artesano de Craft.ec

Una vez completada esta información, un usuario ya puede crear un producto o servicio, para esto se debe dar *click* en el botón que dice “Nueva Publicación” como muestra la Figura 12.



## 5. Crear un producto

Ahora para crear un producto se debe seleccionar el botón “producto” como se muestra en la Figura 13.



A continuación se debe llenar toda la información básica que aparece en el formulario de la Figura 14, poner al menos una etiqueta, y llenar la descripción, en el campo de etiqueta se debe escribir una palabra y seleccionar de las opciones o escribir una y presionar Enter.

The screenshot shows the 'Nuevo producto' form in the Craft.ec interface. At the top, there is a red navigation bar with the user's name 'Rafael', links for 'Publicar un producto' and 'Publicar un servicio', a search bar with the text 'buscar', and a 'Salir' button. On the left, a vertical menu contains links for 'Mi perfil', 'Soy Craft (artesano)', 'Mensajes', 'Nueva publicación', and 'Publicaciones'. The main content area is titled 'Nuevo producto' and contains three input fields: 'Etiqueta/as', 'Nombre', and 'Descripcion'. At the bottom of the form, there is a 'SIGUIENTE' button with a right-pointing arrow.

Figura 14. Formulario de nuevo producto Craft.ec

Después se debe llenar la información del precio en la forma que se muestra en la Figura 15, es obligatorio ingresar el costo unitario.

The screenshot shows the 'Información del precio' section of the 'Nuevo producto' form. The section has a red header with 'Información general' and 'Información del precio'. The main content area is titled 'Información del precio' and contains the following elements: a 'Precio unitario' slider set to '\$134'; a checkbox for 'Es negociable' which is unchecked; a dropdown menu for 'Esta en promoción?' with the selected option 'No hay promoción'; a toggle switch for 'Se realizan envíos?' which is turned on to 'Si'; and a checked checkbox for 'El precio incluye envío'. At the bottom of the section, there are 'SIGUIENTE' and 'ATRÁS' buttons.

Figura 15. Sección de precio de nuevo producto Craft.ec

En esta sección se puede especificar si el producto tiene promoción y también si es que se realiza envíos. Finalmente se debe subir las fotos del producto.

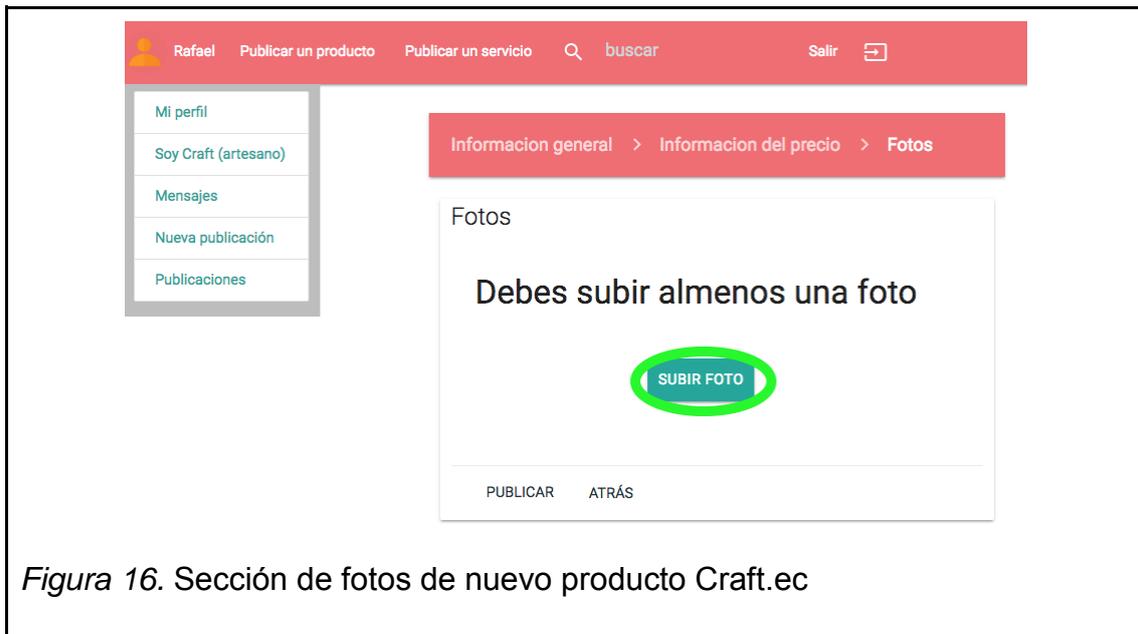


Figura 16. Sección de fotos de nuevo producto Craft.ec

Para subir una foto se debe hacer *click* al botón “Subir Foto” como se muestra en la Figura 16, se selecciona una foto del dispositivo y se debe esperar hasta que la esté visible en la pantalla, se puede subir hasta cinco fotos.

Una de las fotos que el usuario sube va a ser la principal de la publicación, la cual saldrá en la presentación del producto o servicio. Así que para marcar alguna como principal se debe deslizar las imágenes hasta que la foto que esté en el frente y se debe pulsar en el botón “Establecer foto principal” o el botón “eliminar” para borrarla. La Figura 17 muestra la sección de imágenes de la publicación.



Figura 17. Sección de fotos llena para nuevo producto en Craft.ec

Una vez terminado se debe hacer *click* en “Publicar” y se podrá ver todos los productos y servicios creados o también dando *click* en el botón de “Mis publicaciones”

## 6. Crear un Servicio

Para crear un servicio se debe seleccionar el botón “Servicio” desde la sección de “Nueva Publicación” como se muestra en la Figura 18.



Figura 18. Creando nuevo servicio en Craft.ec

A continuación se debe llenar toda la información básica, poner al menos una etiqueta, para esto se debe escribir una palabra y seleccionar de las opciones o presionar Enter a la palabra que el usuario escribió. Para finalizar esta sección se debe hacer *click* en el botón “siguiente” que se muestra en la Figura 19.

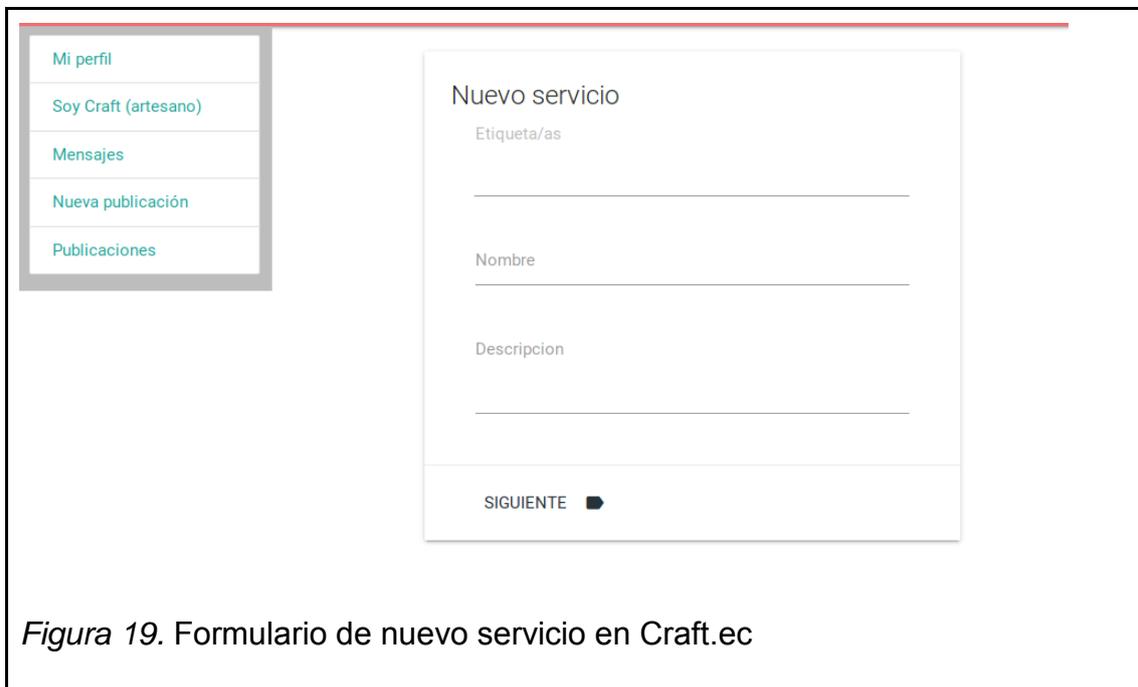


Figura 19. Formulario de nuevo servicio en Craft.ec

Ahora se debe llenar la información acerca del costo del servicio, primero como es el tipo de pago y segundo si es que este es negociable, al final dando *click* a “siguiente” como se muestra en la Figura 20 se continúa hacia la última sección.

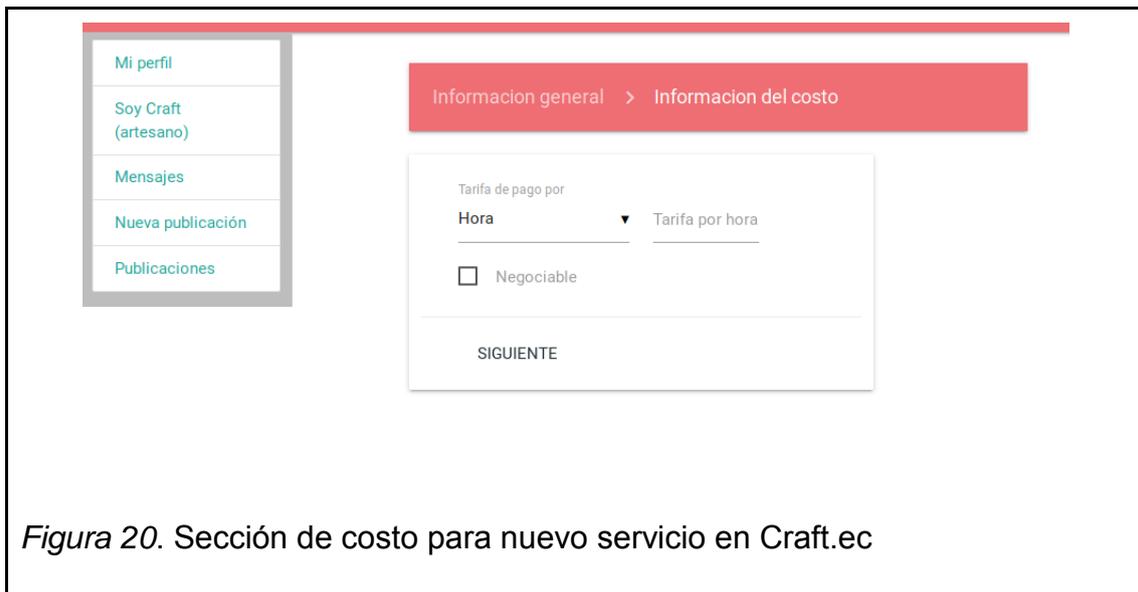


Figura 20. Sección de costo para nuevo servicio en Craft.ec

Finalmente en la última sección, se debe subir una imagen que será la presentación de tu servicio, se puede subir hasta cinco imágenes que respalden el servicio. Al final hay una serie de espacios para que se agregue un enlace hacia una muestra del servicio que ayude a demostrar la calidad del trabajo (Ej. documento, página web, blog, etc.). La Figura 21 muestra la sección de imágenes cuando se publica un servicio.

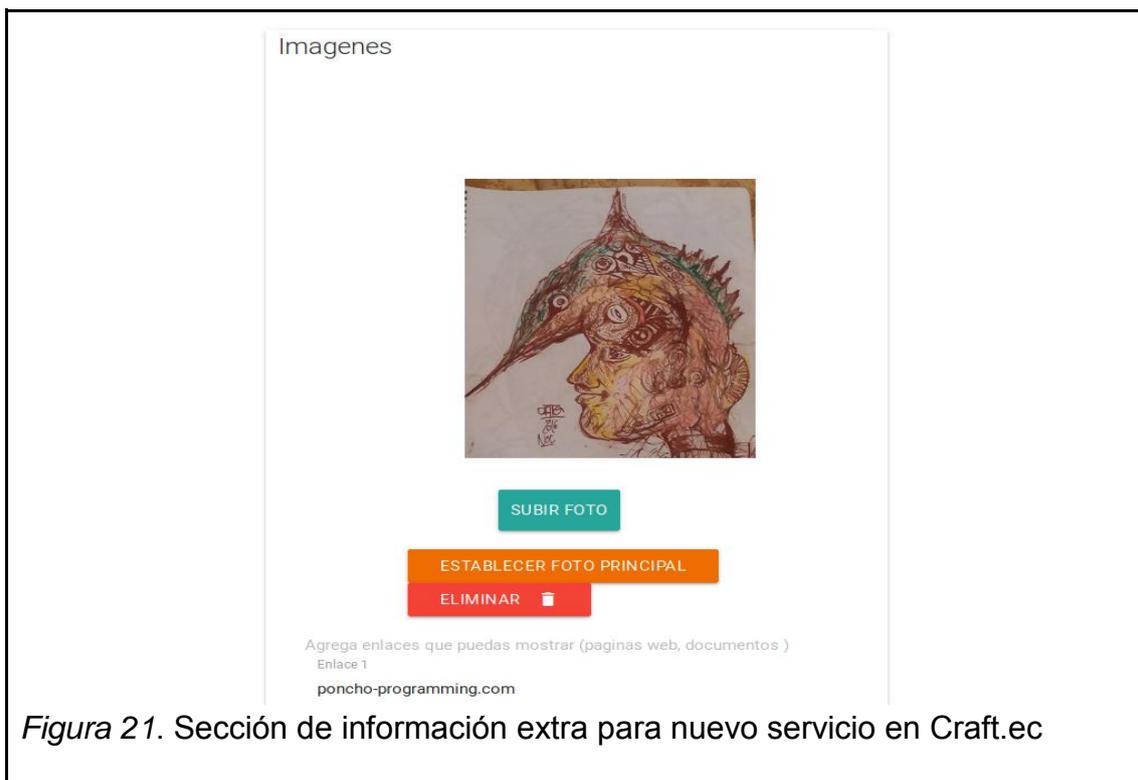


Figura 21. Sección de información extra para nuevo servicio en Craft.ec

Dando *click* en publicar el servicio estará listo para ser encontrado en las búsquedas, y se podrá ver la lista de publicaciones que el usuario tiene actualmente.

## 7. Manejar las publicaciones

Para ver las publicaciones, el usuario desde la pantalla principal, debe hacer *click* en el botón donde está su nombre, después hacer *click* en la opción “Mi cuenta” como se muestra en la Figura 22.

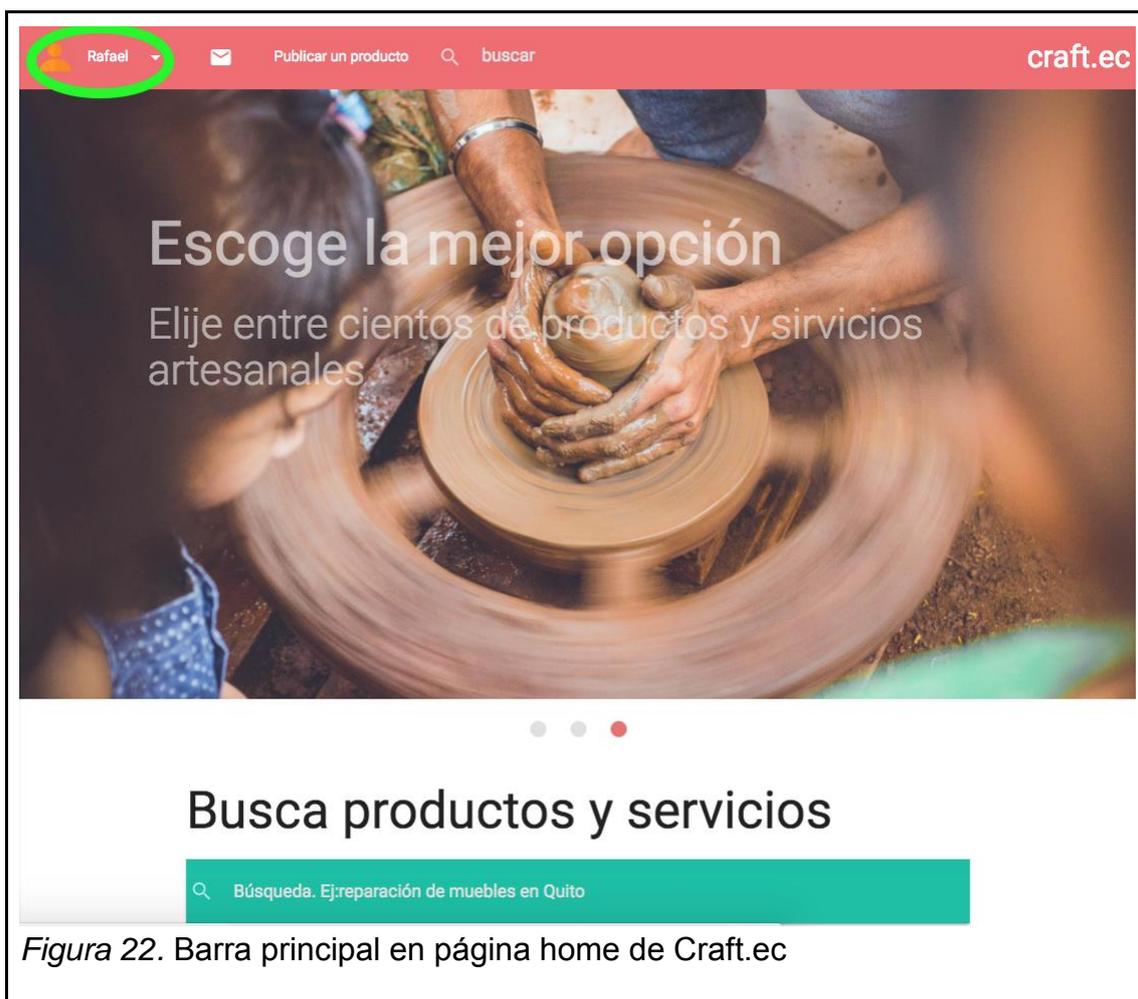


Figura 22. Barra principal en página home de Craft.ec

Una vez en la pantalla de la cuenta del usuario, haciendo *click* en el botón que dice “Mis Publicaciones” en el menú de la izquierda, aparecerán todas las publicaciones que tenga el usuario como se muestra en la Figura 23.

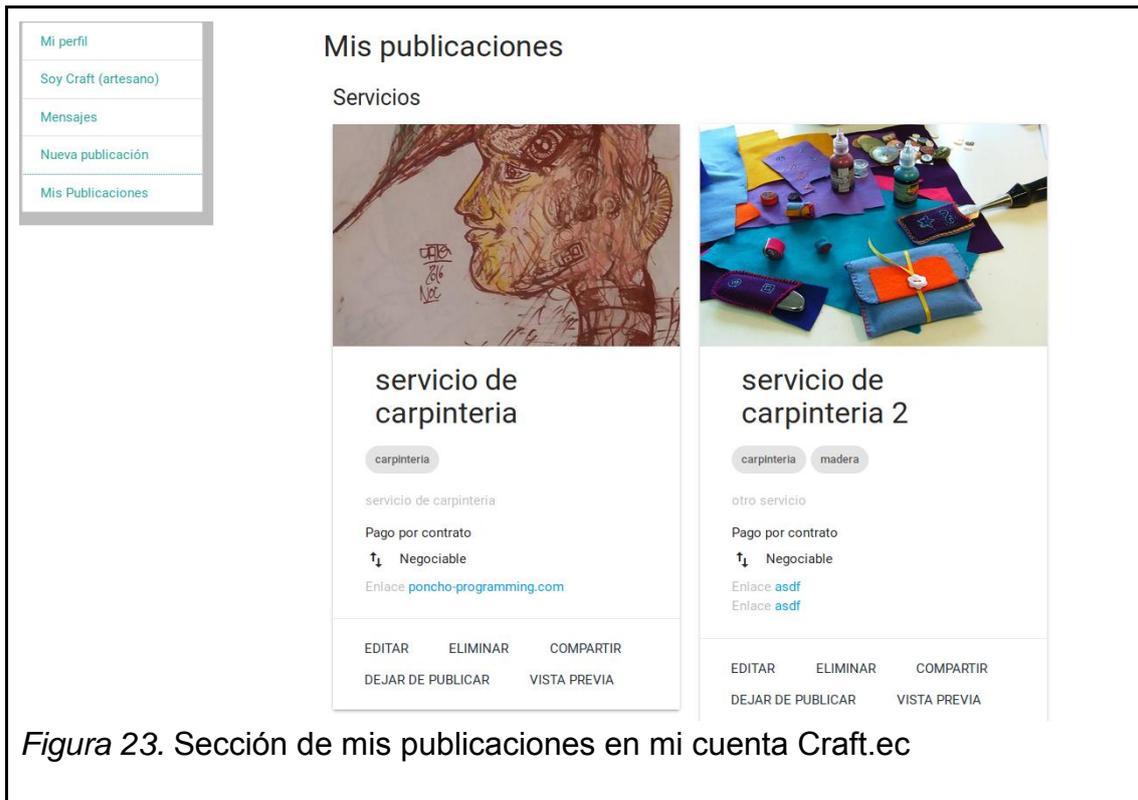


Figura 23. Sección de mis publicaciones en mi cuenta Craft.ec

Desde aquí se puede ver todos los productos y servicios, además de poderlos manejar, Craft.ec permite:

- Editar una publicación.
- Eliminar una publicación.
- Compartir en Facebook.
- Dejar de Publicarla.
- Realizar una vista previa.
- Y volver a publicar en el caso que no esté publicada.

Si no se ha terminado una publicación, al hacer *click* en “Publicar” la aplicación se dirigirá al paso donde se quedó el usuario para pueda publicarla.

## 8. Buscar por un producto o servicio

Desde la página principal de Craft.ec se puede buscar por un producto o servicio simplemente escribiendo la necesidad (Ej. madera en Quito) en la barra de búsqueda como se muestra en la Figura 24.



Figura 24. Barra de búsqueda en la página principal Craft.ec

Al presionar Enter en la barra de búsquedas, se presentará una nueva página con todos los resultados que Craft.ec tiene para ofrecer tal como muestra la Figura 25.

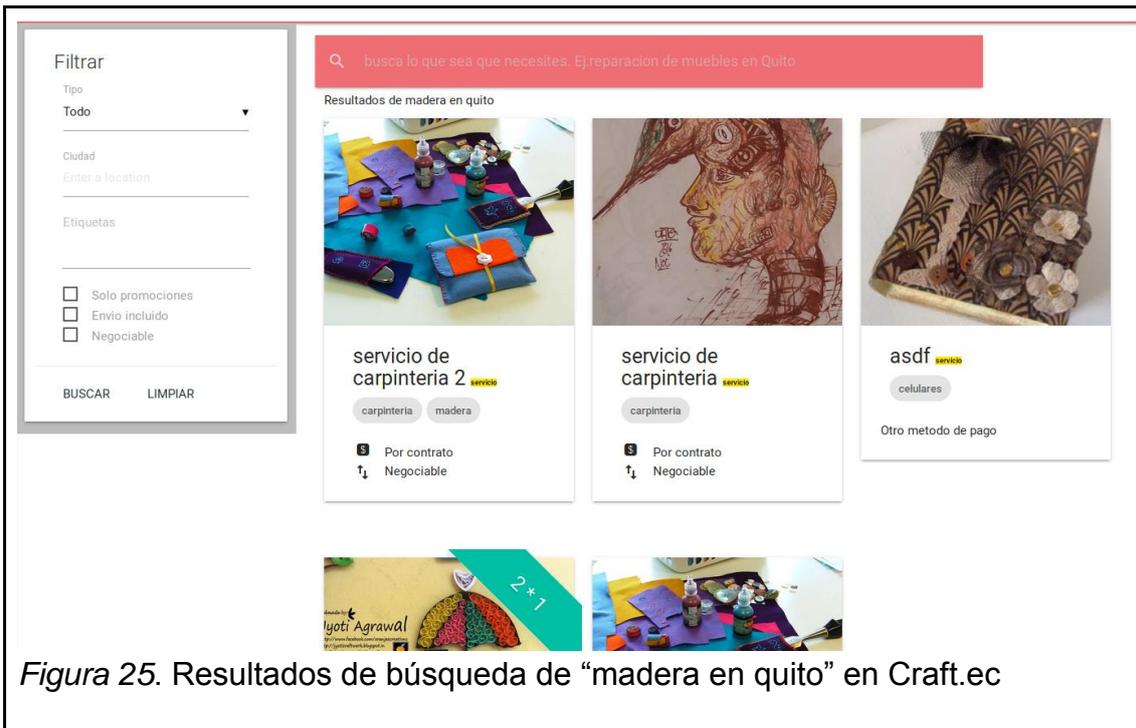


Figura 25. Resultados de búsqueda de "madera en quito" en Craft.ec

A partir de los resultados obtenidos se puede filtrar la lista que aparece en pantalla dependiendo del tipo, de la ciudad etc. Usando la forma que se encuentra al lado izquierda señalada en la Figura 26.

Cada que se use un nuevo filtro en el menú se debe pulsar “Buscar” para que se apliquen a la búsqueda. La página también tiene la barra de búsqueda para que se ingrese un nuevo criterio. Se debe presionar Enter una vez que se haya ingresado una nueva palabra.

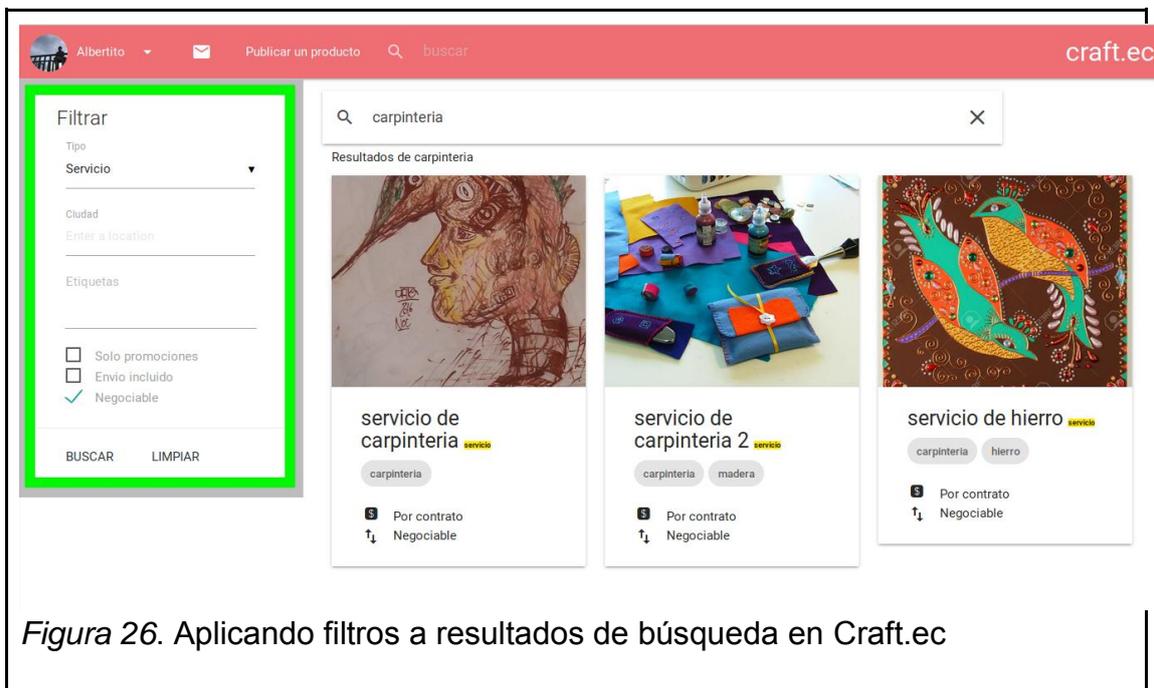


Figura 26. Aplicando filtros a resultados de búsqueda en Craft.ec

Cuando se haya visto la publicación de la que desea más información, únicamente se da *click* sobre ella, esto hará que la aplicación se dirija hacia la información de dicho producto o servicio y su vendedor, como se muestra en la Figura 27.

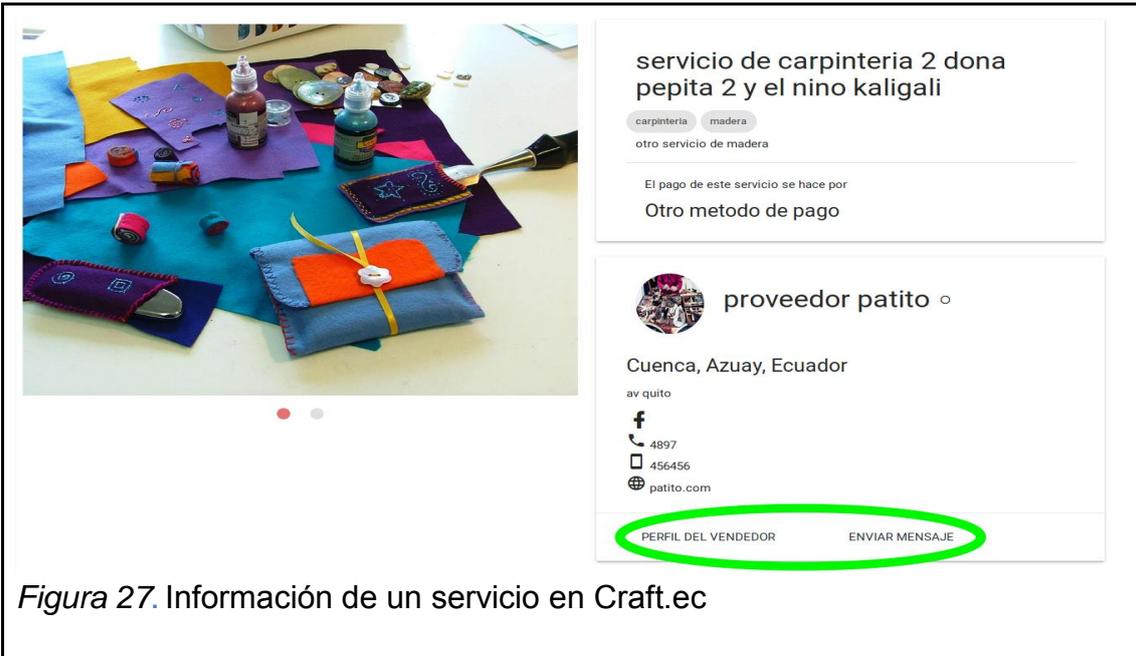


Figura 27. Información de un servicio en Craft.ec

Si se desea ver el perfil del vendedor junto con su información completa, se debe hacer *click* en “Perfil del vendedor”, también se tiene la opción de enviar directamente un mensaje en “Enviar un Mensaje”. La Figura 28 muestra el perfil de un artesano con sus productos y servicios.

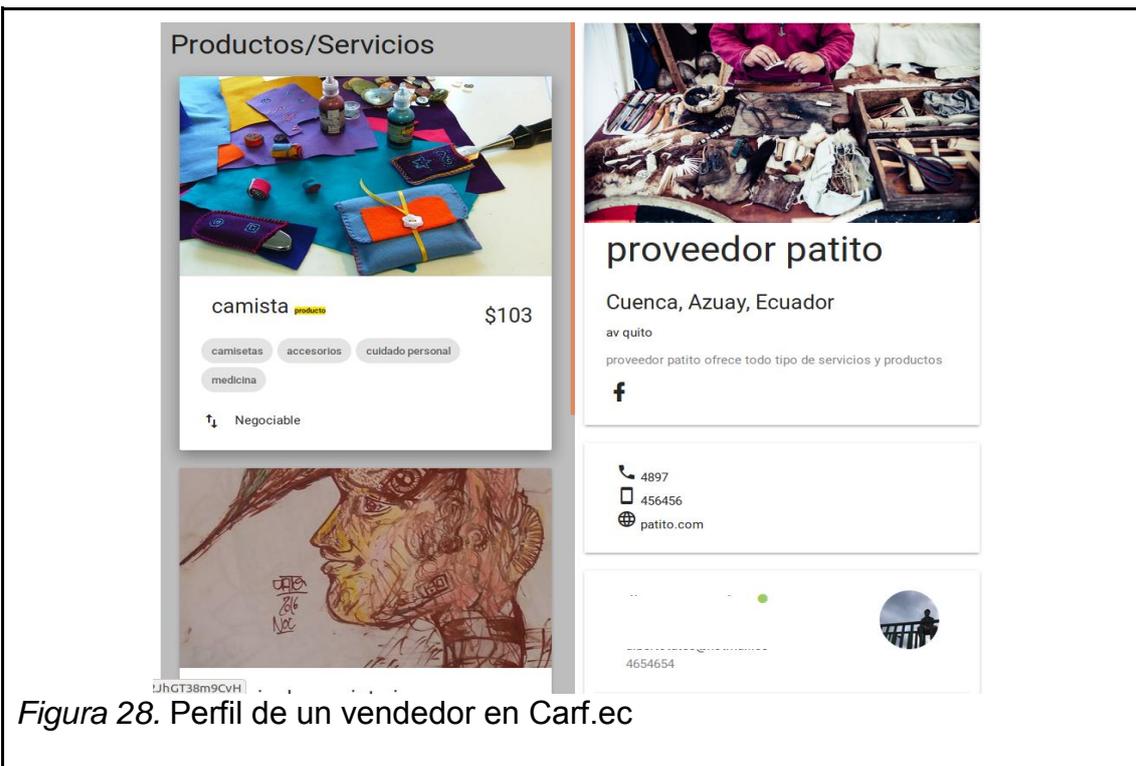


Figura 28. Perfil de un vendedor en Carf.ec

## 9. Enviar un mensaje de chat

A partir del perfil de un artesano, se debe hacer *click* en el botón “Enviar Mensaje” que está marcado en la Figura 28.

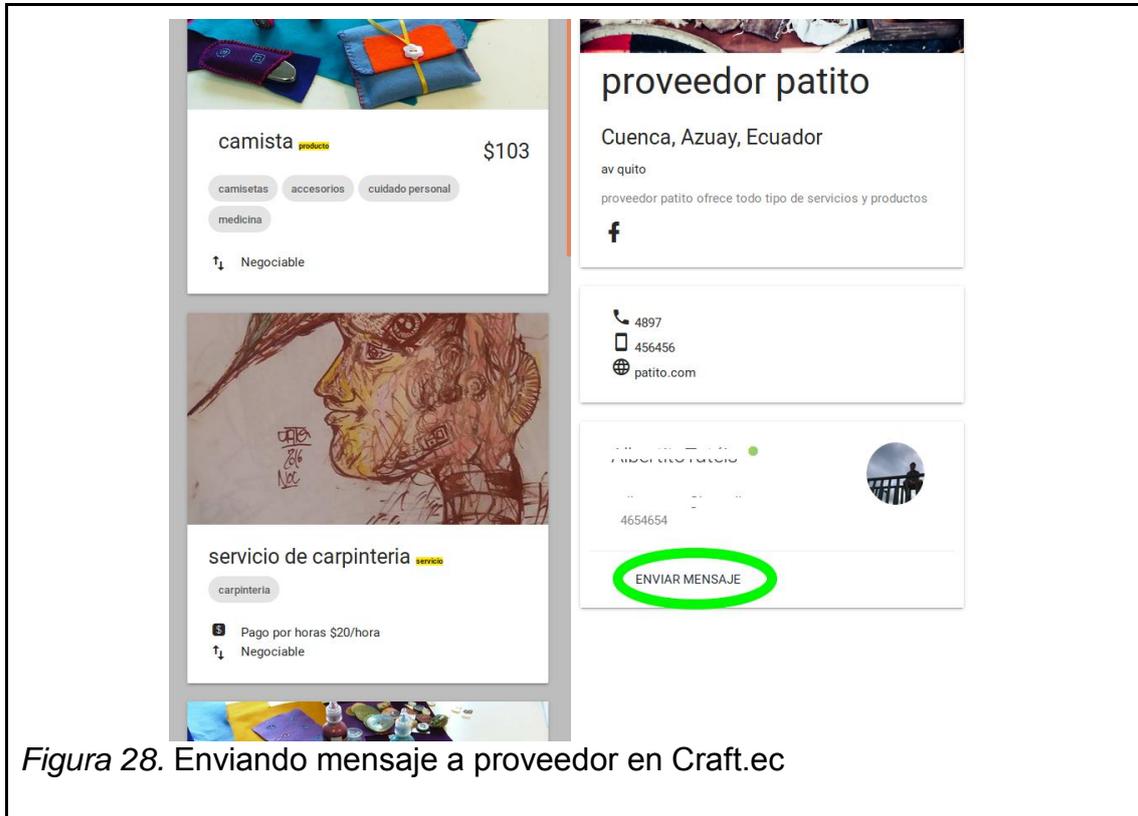


Figura 28. Enviando mensaje a proveedor en Craft.ec

A continuación Craft.ec se redirige a una nueva página, donde se ubican las conversaciones a la izquierda y a la derecha los mensajes con dicho usuario. En el caso de que sea la primera vez, te aparecerá el mensaje de que no tienes conversaciones como se muestra en la Figura 29.

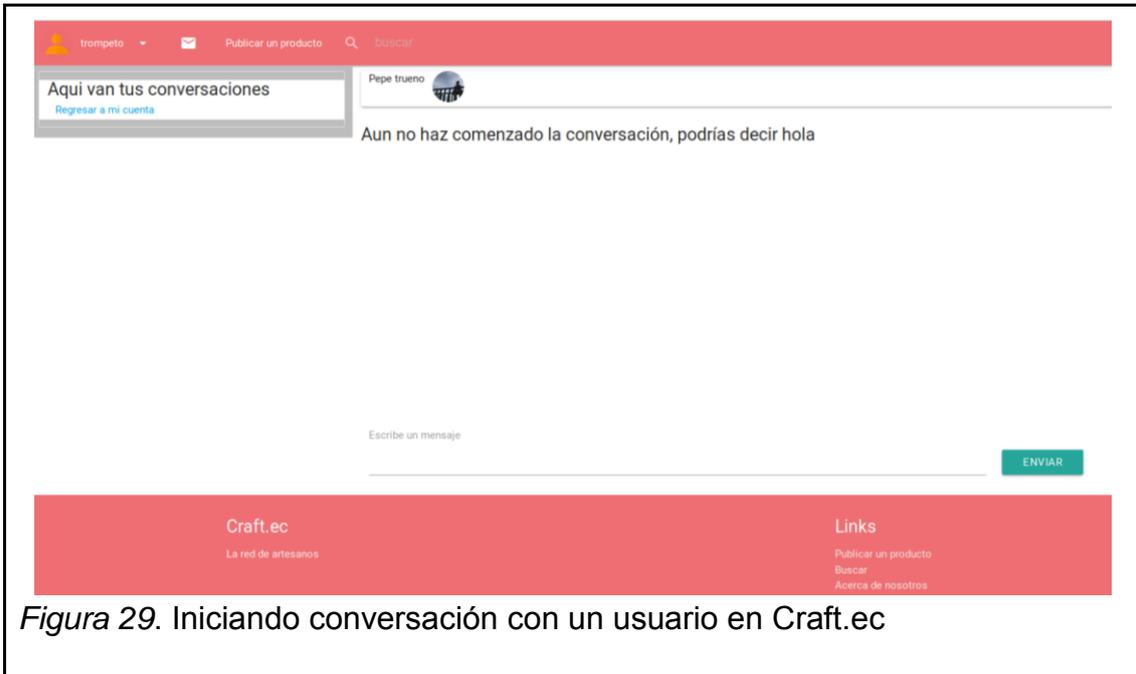


Figura 29. Iniciando conversación con un usuario en Craft.ec

Para que un usuario pueda ver sus mensajes, debe dar *click* en la barra de la aplicación en el icono de correo, o también puede ir a la sección en la página con información del usuario. La Figura 30 muestra la bandeja de un usuario y las conversaciones que ha tenido.

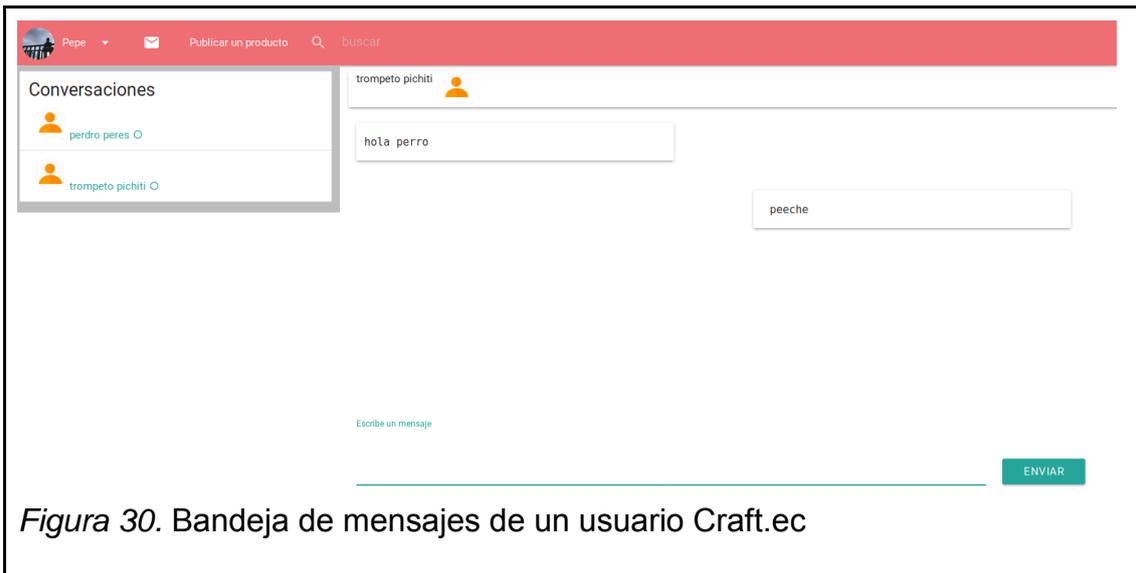


Figura 30. Bandeja de mensajes de un usuario Craft.ec

El icono de correo notifica siempre que el usuario tengas un mensaje nuevo como se marca en la Figura 31.

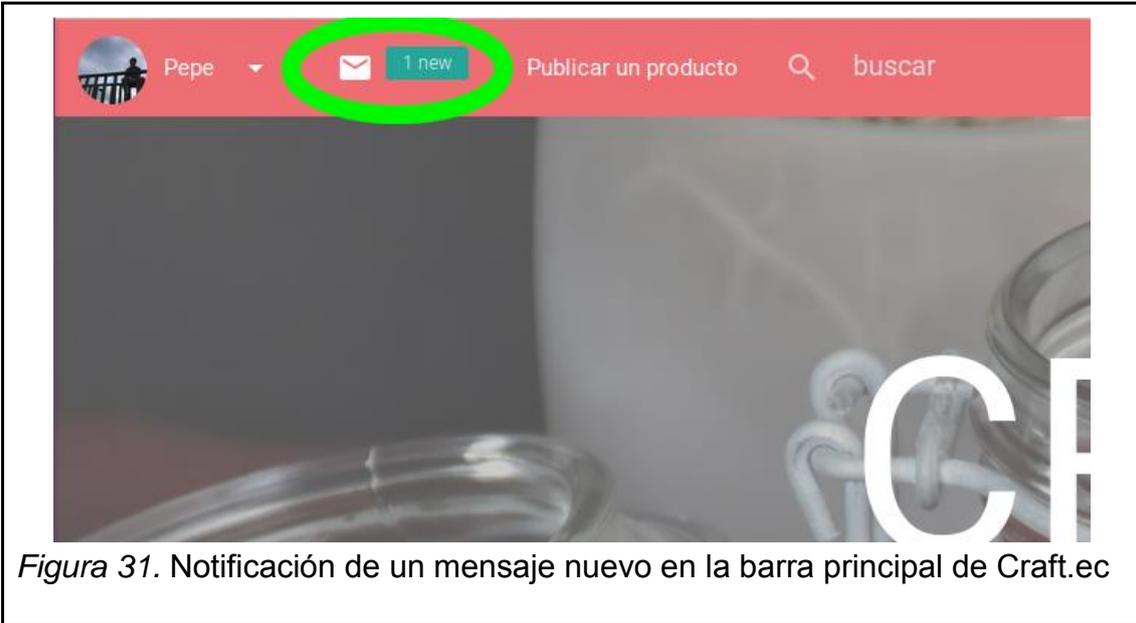


Figura 31. Notificación de un mensaje nuevo en la barra principal de Craft.ec

