

**UNIVERSIDAD NACIONAL DEL NORDESTE**

**FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA**

**Licenciatura en Sistemas de Información**

**Cátedra:** Taller de Programación 1.

**Proyecto:** “***Bruno Gas***”

**Alumno:** Almirón, Diego.

**Año:** 2019.

**Índice:**

[1. INTRODUCCION 5](#_Toc517046298)

[1.1 Breve estado del arte 5](#_Toc517046299)

[1.2 Objetivos 5](#_Toc517046300)

[1.3 Fundamentación 5](#_Toc517046301)

[2. METODOLOGIA 7](#_Toc517046302)

[2.1 Ciclo de Vida 7](#_Toc517046303)

[2.2 Método de educción de requisitos 8](#_Toc517046304)

[2.2.1 Brainstorming 8](#_Toc517046305)

[2.3 Requerimientos Específicos 9](#_Toc517046306)

[2.3.1 Requerimientos funcionales 9](#_Toc517046307)

[1.1.2 Tabla: Requerimientos Funcionales 10](#_Toc517046308)

[2.3.3 Requerimientos no funcionales 11](#_Toc517046309)

[2.4 Análisis de Riesgos 11](#_Toc517046310)

[2.4.1 Tipos de riesgos 12](#_Toc517046311)

[2.4.3 Identificación de Riesgos 12](#_Toc517046312)

[2.4.5 Planificación de riesgos 13](#_Toc517046313)

[2.5 Planificación de actividades 13](#_Toc517046314)

[2.5.1 Planificación de Tareas 13](#_Toc517046315)

[2.5.4 Estimación de costos 15](#_Toc517046316)

[2.5.5 Resultados en Cocomo 2 17](#_Toc517046317)

[2.5.7 Resultados en Cocomo 2 18](#_Toc517046318)

[3. HERRAMIENTAS Y/O LENGUAJES DE PROGRAMACION 19](#_Toc517046319)

[3.1 Lenguajes de programación 19](#_Toc517046320)

[3.1.1 Lenguaje HTML5 19](#_Toc517046321)

[3.1.2 Lenguaje PHP 19](#_Toc517046322)

[3.1.3 Lenguaje SQL 20](#_Toc517046323)

[3.1.4 lenguaje CSS3 20](#_Toc517046324)

[3.2 Herramientas 20](#_Toc517046325)

[3.2.1 Herramienta Project libre 20](#_Toc517046326)

[3.2.2 Herramienta Cocomo 2 21](#_Toc517046327)

[3.2.3 Herramienta Enterprise Architect 21](#_Toc517046328)

[3.2.4 Herramienta NDT Suite 22](#_Toc517046329)

[4. RESULTADOS 23](#_Toc517046330)

[4.1 Resultados obtenidos 23](#_Toc517046331)

[4.1.1 Caso de uso 24](#_Toc517046332)

[4.1.3 Entidades y Relaciones 25](#_Toc517046334)

[4.2 Resultados NDT 26](#_Toc517046335)

[4.2.1 Participantes 26](#_Toc517046336)

[4.2.3 Actores 26](#_Toc517046337)

[4.2.5 Objetivos 28](#_Toc517046339)

[4.2.7 Navegación 28](#_Toc517046339)

[4.3 Prototipos 29](#_Toc517046341)

[4.3.1 Página Principal 30](#_Toc517046342)

[4.3.3 Login y Registro 31](#_Toc517046343)

[4.3.5 Catalogo 32](#_Toc517046344)

[4.3.7 Consultas 33](#_Toc517046345)

[5. CONCLUSIONES 34](#_Toc517046348)

[5.1 Referencias 34](#_Toc517046349)

**Índice de figuras:**

2.5.2 Figura: Diagrama Gantt del sistema 13

2.5.3 Figura: Tareas a Realizar 13

2.5.6 Figura: Resultados en Cocomo 2 16

2.5.8 Figura: Resultados en Cocomo 2 17

4.1.2 Figura: Diagrama de Caso de uso 22

4.1.4 Figura: Diagrama de Entidad Relación 23

4.2.2 Figura: Diagrama de Participantes 24

4.2.4 Figura: Diagrama de Actores 24

4.2.6 Figura: Diagrama de Objetivos 26

4.2.8 Figura: Diagrama de Navegación 26

4.3.2 Figura: Página principal del Sistema 28

4.3.4 Figura: Iniciar Sesión o Identificarse 29

4.3.6 Figura: Catálogo de Productos 30

4.3.8 Figura: Consultas 31

**Índice de tablas:**

2.2.2 Tabla: Tormenta de ideas del proyecto……………………………………………………………………………8

2.3.2 Tabla: Requerimientos Funcionales…………………………………………………………………………………9

2.3.4 Tabla: Requerimientos No Funcionales 10

2.4.2 Tabla: Tipos de riesgos…………………………………………………………………………………………………….10

2.4.6 Tabla: Planificación de Riesgos 11

2.4.8 Tabla: Identificación de Riesgos 11

# 

# **1.** INTRODUCCION

El presente documento es una Especificación de Requisitos de Software perteneciente al desarrollo de un sistema para la Venta y Stock de “Bruno Gas”. También se pretende administrar los productos y que los clientes puedan administrar sus perfiles de usuario. En cuanto a las ventas se pretende obtener datos estadísticos para que el negocio pueda llevar a cabo un estudio en cuanto a ganancias y crecimiento del negocio.

## **Breve estado del arte**

Debido a la facilidad y rapidez del manejo de datos de las nuevas tecnologías en el manejo de la sociedad en general, han surgido nuevas necesidades de parte de los comerciantes. Entre dichas carencias está la automatización de procesos de venta y control de las mercaderías, en este apartado se deben tomar en cuenta varios aspectos como: capacidad para gestionar los stocks de productos, así como también la realización de altas bajas y modificaciones de dichos productos, facilidad en el control y los números de las ventas realizadas. También se ofrecerá seguridad y confiabilidad al vender un producto, y mejorar la atención al público. Por esta razón la demanda por sistemas que cubran estos requisitos es latente en los comercios hoy día. De esta manera se plantea una solución informática para cubrir los aspectos anteriormente señalados.

## 1.2 Objetivos

Se establece como objetivo la realización de un Sistema de Gestión de Productos perteneciente a una organización que permita las ventas informatizadas y la promoción de los mismos. Que la información se realice en forma clara y precisa, discriminando cada publicación según sus categorías.

## 1.3 Fundamentación

Tras analizar la forma en que la “Bruno Gas” gestionaba y manejaba la venta y promoción de sus productos, se ha detectado un número importante de falencias entre las cuales podemos mencionar; incapacidad para controlar el manejo de mercaderías, dificultad en la generación de informes de los ingresos, entre otros. Estos eran llevados a cabo de forma manual y no garantizaba la persistencia de dicha información.

# 2. METODOLOGIA

## 2.1 Ciclo de Vida

Para este proyecto trabajaremos con la metodología NDT, que permite identificar los objetivos, las alternativas y las restricciones. Tendremos en cuenta lo siguiente:

**Planificación:**

* Se desarrollara un software de gestión para una tienda en línea, para poder normalizar el registro de ventas, búsquedas y solicitud de pedidos de productos.

**Análisis:**

* El sistema permitirá cargar los datos de clientes y productos, que serán guardados en una base de datos, así como también realizar la búsqueda y encontrar faltantes de los mismos.

**Diseño:**

* Se diseñará un sistema sencillo que sea intuitivo, y de fácil manejo, la maquina donde se instalará el sistema no requerirá de muchos recursos.

**Codificación:**

* Esta etapa abarca como estará organizada la información en el sistema, en este caso será a través de un código de identificación de cada producto. La búsqueda se podrá hacer a través del categorías.

**Implementación:**

* Se utilizará entorno Codeigniter como software para el diseño del sistema.

**Mantenimiento:**

* La mantención se hará de forma semanal a través de backup de datos, y si el cliente lo requiere, se harán modificaciones al sistema siempre y cuando sean transformaciones que no requieran cambiar tanto el código, para mantener actualizado el software.

En nuestro caso usaremos el modelo NDT, ya que es el que mejor se adapta a los requerimientos estipulados para el desarrollo de esta aplicación en particular.

## 2.2 Método de educción de requisitos

El método de educción de requisitos seleccionado para el desarrollo del sistema es tormenta (Brainstorming) de ideas ya que consideramos muy factible para este tipo de sistemas que administra mercaderías. Para ello se solicita la presencia del cliente y el equipo que llevara a cabo el desarrollo del sistema.

### 2.2.1 Brainstorming

|  |
| --- |
| **Fecha: 18/06/2018 Hora: 19:00 Duración: 3 horas** |
| **Asistentes:**  Cliente, Equipo de Desarrollo |
| **Objeto de la sesión:**  Obtener una idea de los requisitos para construir los requisitos del sistema |
| **Ideas aportadas:**  1) Que el sistema registre los clientes que deseen comprar productos  2) Que el sistema registre los productos  3) Que el sistema permita buscar productos por categoría  4) Que el sistema permita buscar productos por mayor precio  5) Que el sistema permita buscar productos por menor precio  6) Que el sistema registre los distintos tipos de productos que tendrá el sistema  7) Que el sistema calcule el costo total de los productos que se deseen comprar  8) Que el sistema admita distintos tipos de pagos con sus descuentos  9) Que el sistema registre las compras realizados por los clientes  10) Que el sistema proponga estrategias de publicidad y promociones  11) Que el sistema permita agregar nuevos servicios para ampliar el sistema  12) Que el sistema permita administrar los productos con un sistema de alta, baja y modificación.  13) Que el sistema permita que un usuario pueda administrar su perfil con un sistema de alta y modificación  14) Que el sistema calcule estadísticas que faciliten la toma de decisiones para mejorar el servicio de atención al cliente  15) Que el sistema permita una adaptación multiplataforma para poder adaptarse tanto como PC, tabletas y dispositivos móviles  16) Que el sistema presente una interfaz gráfica llamativa y amigable para interactuar con el cliente |

**2.2.2 Tabla: Tormenta de ideas del proyecto**

## 2.3 Requerimientos Específicos

### 2.3.1 Requerimientos funcionales

|  |  |
| --- | --- |
| **REQUERIMIENTO FUNCIONAL** | **DESCRIPCIÓN** |
| RF-01: Realizar el alta de un nuevo cliente ( apellido, nombre, contraseña, domicilio, correo y teléfono). | El sistema debe permitir que un nuevo usuario pueda registrarse en el sistema suministrando todos los datos que se le pida. |
| RF-02: Realizar la modificación de los usuarios (apellido, nombre, contraseña, domicilio, correo y teléfono). | El sistema debe posibilitar al administrador y al cliente del sistema a modificar los datos en caso de que fuera necesario. |
| RF-03: Realizar el control de stock de los Productos. | El sistema implementara el carrito de compras en donde se irán cargando los productos a comprar y establecerá el precio final de los mismos. |
| RF-04: Imprimir facturas y recibo | El sistema debe imprimir en una factura con diseño estándar, los datos registrados durante la venta de lo/s producto/s. En dicho diseño se deberá incluir: Datos del cliente: Apellido, Nombre, Domicilio, y Teléfono. Datos de la venta: forma de pago, la fecha e importe final. |
| RF-05: Implementar una base de datos correspondiente al total de ventas efectuadas dentro del sistema. | El sistema debe permitir la administración de la totalidad de ventas registradas. En la misma se almacenarán: los productos vendidos, la fecha y hora de los mismos. |
| RF-06: Realizar el alta de un nuevo producto( descripción, nombre, stock y precio.). | El sistema debe permitir que se pueda agregar al sistema un nuevo producto producto con sus respectivas descripciones. |
| RF-07: Realizar la baja de un producto del sistema. | El sistema debe permitir que se pueda eliminar un producto del sistema si fuera necesario, el cual se eliminaría de la base de datos. |
| RF-08: Realizar la modificación de un producto (nombre, descripción, stock y precio.). | El sistema debe permitir que el administrador pueda modificar los datos de un producto y guardar. |
| RF-09: Que los usuarios (registrados, no registrados) puedan visitar el catálogo de productos. | El sistema debe permitir tanto a los usuarios registrados, como a los no registrados que puedan visitar el catálogo de productos disponibles. |
|  |  |

### 2.3.2 Tabla: Requerimientos Funcionales

### 2.3.3 Requerimientos no funcionales

|  |  |
| --- | --- |
| **CLASIFICACIÓN** | **DESCRIPCIÓN** |
| RNF-01: Organización/Implementación | El sistema se implementará en lenguaje Html5, CSS3 y PHP. |
| RNF-02: Organización/Implementación | El sistema utilizará una base de datos implementada en SQL. |
| RNF-03:Producto/Mantenimiento | El sistema dispondrá de un manual usuario para facilitar el mantenimiento. |
| RNF-04: Producto/Eficiencia | El sistema deberá permitir la adaptación a los sistemas operativos Windows y Android. |
| RNF-05: Organización/Estándar | El sistema se deberá realizar bajo la especificación de requerimientos de software según la norma IEEE 830. |
| RNF-06: Organización/Entrega | El prototipo del sistema se entregará en 22 semanas. |
| RNF-07: Producto/Usabilidad | El sistema deberá estar disponible en horario comercial, todos los días laborales. |
| RNF-08: Producto/Eficiencia | El sistema se recuperará automáticamente tras producirse un fallo. |
| RFN-09: Producto/Rendimiento | El sistema deberá estar preparado para aceptar más de 100 sesiones de clientes al mismo tiempo sin degradar su rendimiento más de un 50%. |

**2.3.4 Tabla: Requerimientos No Funcionales**

## 2.4 Análisis de Riesgos

Durante la etapa análisis hemos detectados los posibles riesgos que pasamos a detallar a continuación:

### 2.4.1 Tipos de riesgos



**2.4.2 Tabla: Tipos de riesgos**

### 2.4.3 Identificación de Riesgos



**2.4.4 Tabla: Identificación de Riesgos**

### 2.4.5 Planificación de riesgos



**2.4.6 Tabla: Planificación de Riesgos**

## 

## 2.5 Planificación de actividades

### 2.5.1 Planificación de Tareas

**Análisis - 3 semanas**

* Entrevista.
* Relevamiento de Funciones actuales del Sistema
* Definir técnica de educción.

**Diseño – 5 semanas**

* Generación ambiente de desarrollo.
* Modelado del Sistema.
* Diseño.

**Desarrollo – 6 semanas**

* Desarrollo de funcionalidad
* Codificación
* Testeo parcial

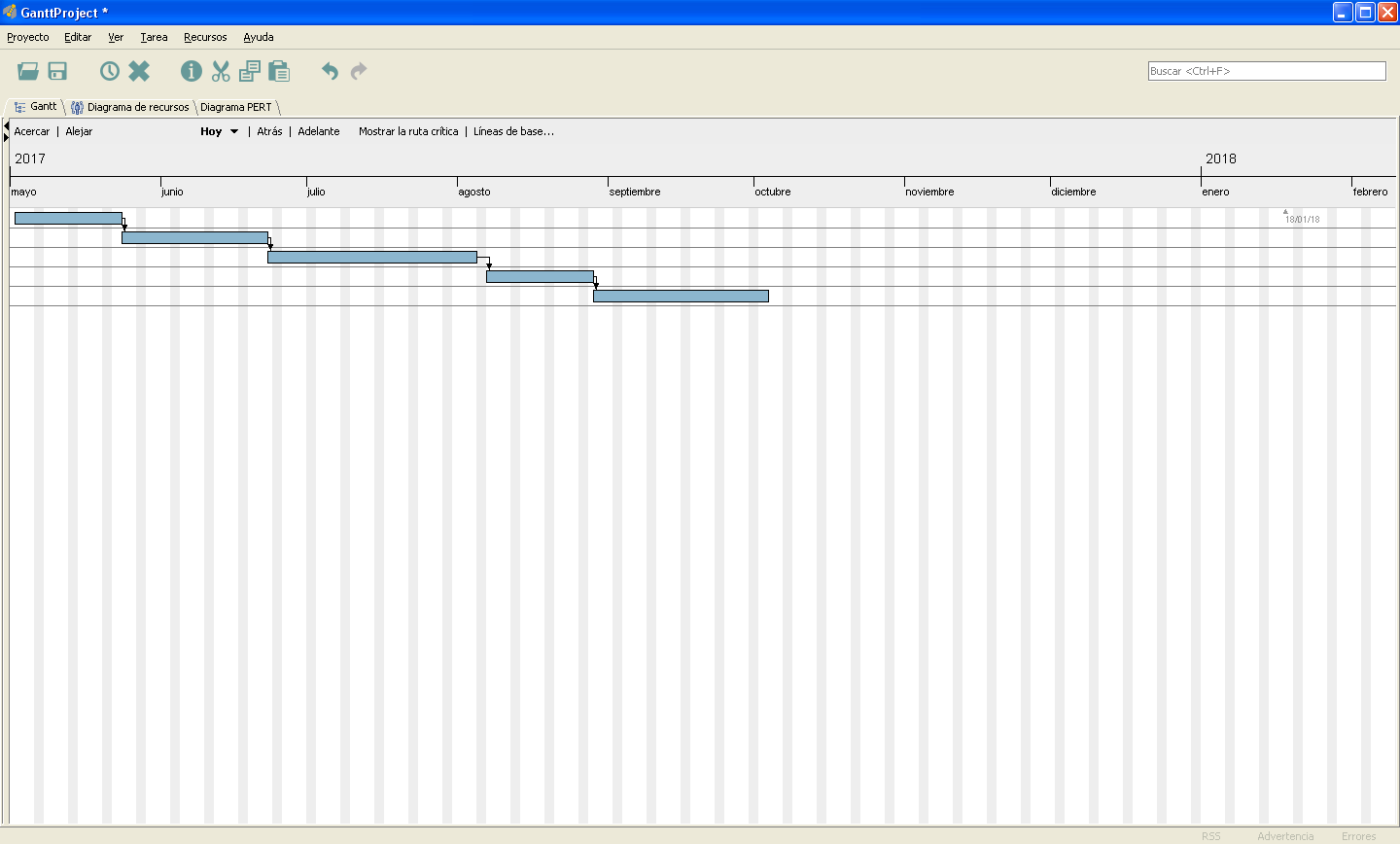
***Primer entrega de la aplicación y consulta de aprobación***

**Prueba – 3 semanas**

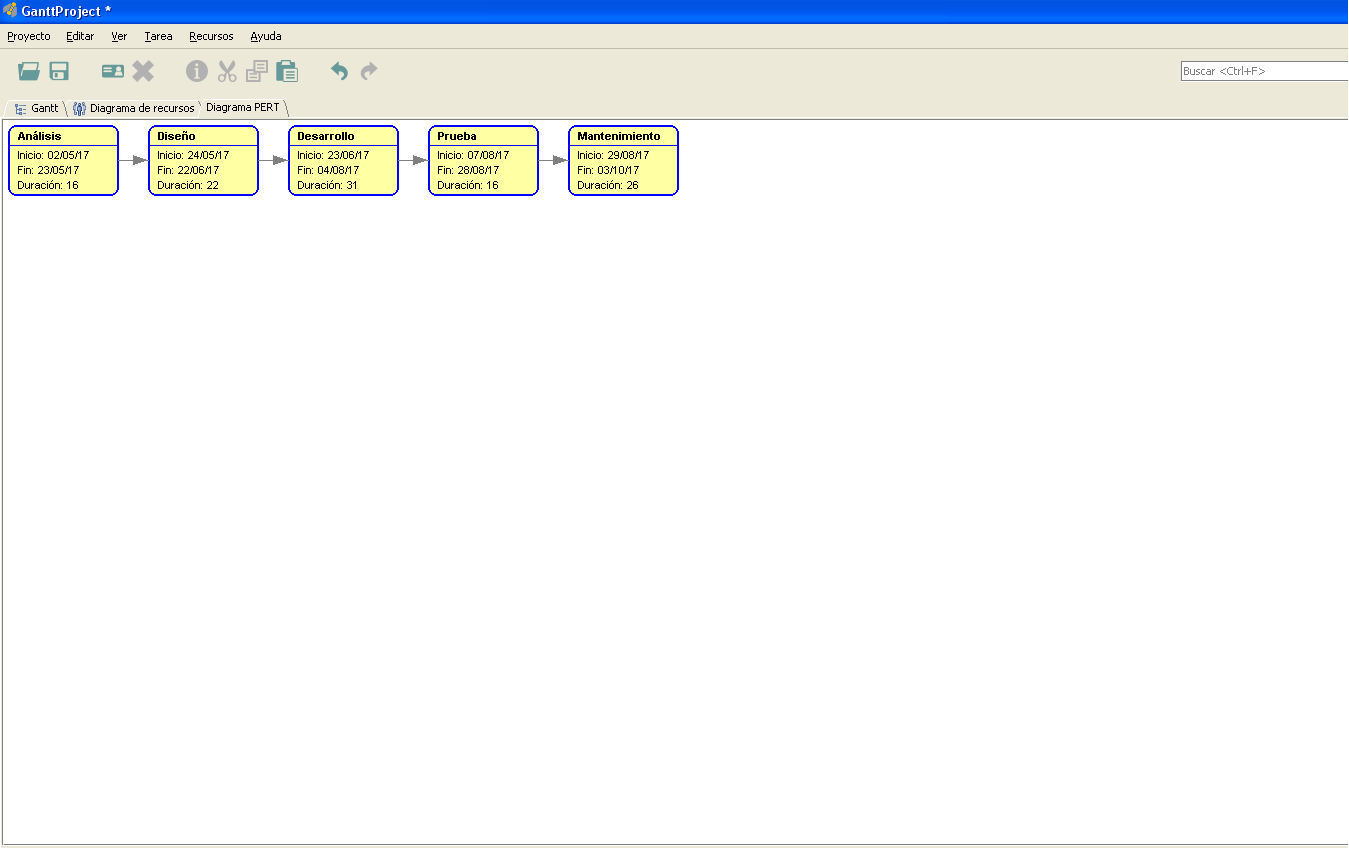
* Prueba del cliente.
* Testeo y depuración.
* Pruebas finales sobre datos reales.

**Mantenimiento – 5 semanas**

* Instalación y puesta en marcha.
* Capacitación.
* Detección y corrección de errores.
* Documentación.



**2.5.2 Figura: Diagrama Gantt del sistema**



**2.5.3 Figura: Tareas a Realizar**

### 2.5.4 Estimación de costos

El método que se ha seleccionado para estimar los costos de este proyecto es el de Cocomo II, que nos servirá para poder obtener una estimación del esfuerzo, la duración, y la cantidad de personas que requerirá el proyecto.

Este sistema que vamos a desarrollar es considerado de tipo orgánico, ya que se estima que tendrá alrededor de 2000 líneas de código.

El método se estimación de costos Cocomo II se obtiene aplicando la siguiente fórmula:

E = A \*(Size B) \* EM.

Donde E corresponde a la estimación del esfuerzo.

A = 2,94.

Size corresponde al tamaño del software, que en este caso es medido en kLCD (Miles de líneas de código).

Entonces:

Size = 2000.

EM =

EM está dado por el producto de los multiplicadores de esfuerzo, de los cuales se considerara los siguientes:

**Personal:**

La capacidad y habilidad de los que vamos a desarrollar el sistema es normal: (ACAP/NOM/1) y (PCAP/NOM/1).

La experiencia con el lenguaje de programación es baja: (LTEX/LO/1,09)

La experiencia con respecto a la plataforma y la aplicación es considerada normal: (PEXP/NOM/1) y (AEXP/NOM/1).

**Producto:**

La confiabilidad requerida es alta: (RELY/HI/1,10).

El tamaño de la base de datos es normal: (DATA/NOM/1).

La complejidad del producto es nominal: (CPLX/HI/1,17).

Los demás multiplicadores de esfuerzo son normales: (RUSE/NOM/1) y (DOCU/NOM/1).

**Plataforma:**

El uso del almacenamiento es normal, al igual que el uso del tiempo de ejecución disponible: (STOR/NOM/1) y (TIME/NOM/1).

El cambio de plataforma es bajo: (PVOL/LO/0.87).

**Proyecto**:

Los atributos que corresponden al proyecto tienen un valor nominal: (TOOL/NOM/1) y (SITE/NOM/1).

B = 1.01 + 0.01 x ∑5i=1 Sf1,

Donde Sf hace referencia a los factores de escala. Los cuales se consideraran de la siguiente manera:

TEAM: el equipo de proyecto estará conformado por 5 personas, cuyas interacciones son altamente cooperativas: (TEAM/1,10).

PREC: se está familiarizado con este tipo de aplicaciones: (PREC/2,48):

FLEX: existe una necesidad muy alta de que la aplicación cumpla con los requisitos pre-establecidos: (FLEX/6.20).

RESL: este factor es considerado normal: (RESL/2.24).

Entonces tenemos que:

A = 2,94

SIZE = 2000 /1000 = 2

B = 1,01 + 0,01 \* (1,10 + 2,48 + 6,20 + 2,24) = 1,130

EM = 1,09 \* 1.10 \* 1,17 \* 0,87 = 1,220

Por lo tanto:

E = 2,94 \* (2 1,130) \* 1,220 = 7,850 ≡ 8 P/M

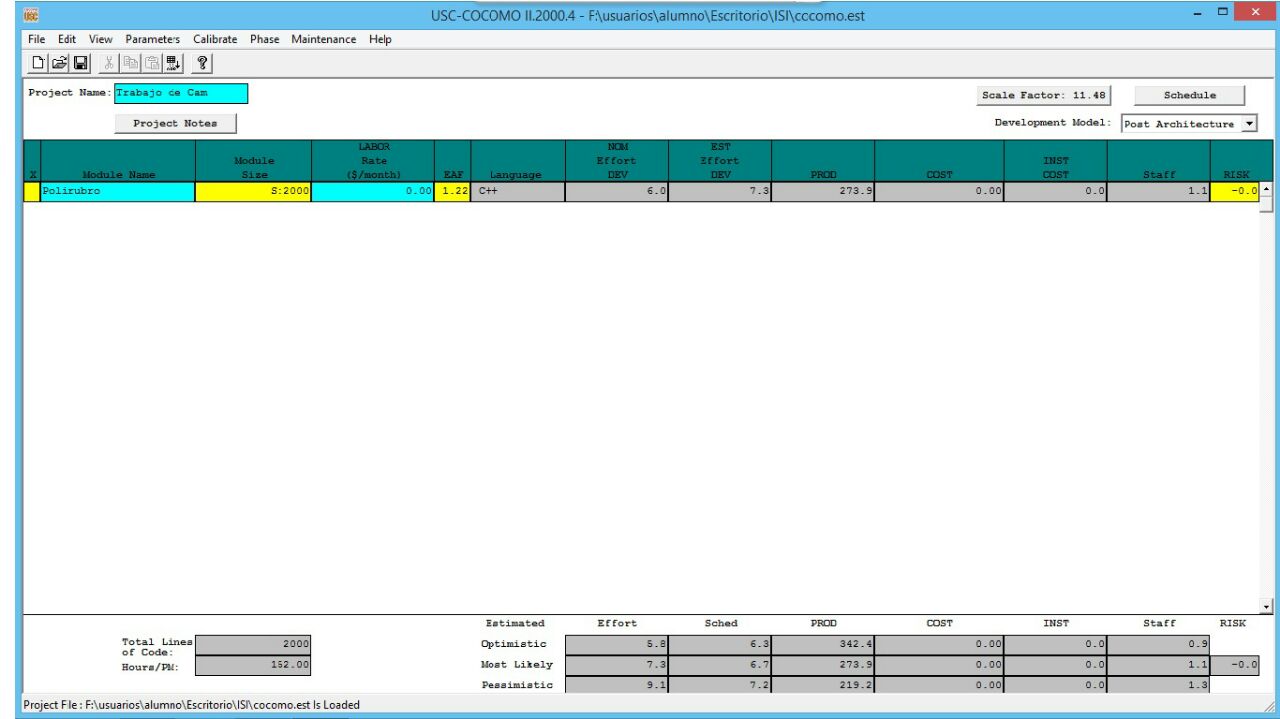
El tiempo o duración del proyecto es:

TDVE = (3,0 \* 8 0,33 + 0,2\*(1,130 - 1,01)) = 3,0 \* 8 0,354 = 6,263 ≡ 7 meses

Y la cantidad de personas es:

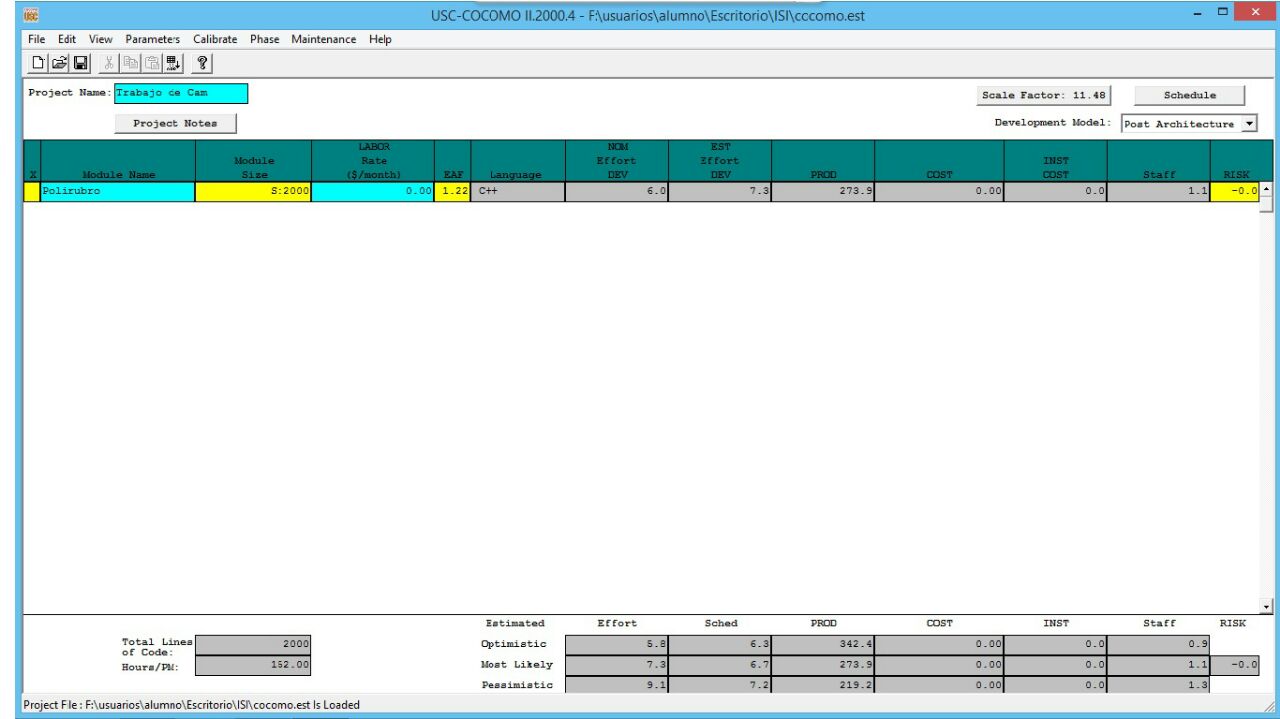
P = 8/7 = 1,142 ≡ 2 personas

### 2.5.5 Resultados en Cocomo 2



**2.5.6 Figura: Resultados en Cocomo 2**

### 2.5.7 Resultados en Cocomo 2



**2.5.8 Figura: Resultados en Cocomo 2**

# 3. HERRAMIENTAS Y/O LENGUAJES DE PROGRAMACION

## 3.1 Lenguajes de programación

### 3.1.1 Lenguaje HTML5

Definiéndolo de forma sencilla, “HTML es lo que se utiliza para crear la mayoría las páginas web de Internet”. Más concretamente, HTML es el lenguaje con el que se “escriben” estas páginas web.

Los diseñadores utilizan el lenguaje HTML para crear sus páginas web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido HTML.

Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas.

### 3.1.2 Lenguaje PHP

Con PHP se puede hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos particulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales ecommerce, donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse MySQL, Oracle, ProgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil, aunque se puede hacer mucho más con PHP. Sus tags van incluidos dentro del código HTML. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades.

### 3.1.3 Lenguaje SQL

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales.

Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales**.**

### 3.1.4 lenguaje CSS3

CSS3 es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS3 es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”).

## 3.2 Herramientas

### 3.2.1 Herramienta Project libre

Con ProjectLibre se pueden realizar diagramas de Gantt y Gráficos Pert; además de diagramas RDBS de estructura analítica de recursos, y diagramas WBS, de estructura de descomposición del trabajo. Su interfaz de usuario, las pestañas de navegación y el procedimiento que emplea para construir un proyecto resultan familiares a las personas que han trabajado en anteriores ocasiones anteriores con Microsoft Project.

Para crear un diagrama de Gantt, lo primero que hay que hacer es redactar una lista de tareas o actividades, desglosando en cada una de ellas también las sub tareas. El siguiente paso consiste en marcar sus fechas de inicio o fin, la duración de cada una de ellas y el orden de prioridades para su ejecución. Aquí también es necesario ver cuales dependen de otras y ver como se vinculan entre ellas. Finalmente se asignan los recursos de forma detallada y para cada una de las actividades definidas.

### 3.2.2 Herramienta Cocomo 2

Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño. Representa el más extenso modelo empírico para la estimación de software.

### 3.2.3 Herramienta Enterprise Architect

Enterprise Architect es renombrado por su enriquecida lista de características. Algunas de ellas están re saltadas en la siguiente lista:

• Comprensivo soporte UML 2.1 para los 13 diagramas UML

• Modelado Extendido para requisitos, diseño de la interfaz de usuario, mapeo de ideas, modelado de datos y más.

• Velocidad: Enterprise Architect es rápido para cargar y un actor espectacularmente rápido, inclusive con grandes modelos.

• Escalabilidad: Enterprise Architect soporta usuarios simples y el desarrollo de pequeños modelos, o muchos usuarios concurrentes desarrollando modelos extremadamente grandes, con igual facilidad.

• Usabilidad: muchos de nuestros usuarios concuerdan, Enterprise Architect lo deja iniciar y producir rápidamente, con una interfaz de usuario enriquecida, y la habilidad de crear patrones, plantillas, vistas de modelo y colecciones “favoritas” de los elementos y diagramas más comúnmente.

• La Administración de Requisitos integrada le permite especificar, trazar y verificar requisitos directamente contra el diseño.

• Extensivo soporte para la administración del proyecto, incluyendo recursos, métricas, definición de pruebas, seguimiento y administración.

• Auditoria de proyecto, líneas base y utilidades de comparación que le permiten registrar y monitorear cambios y (con comparación) volver los cambios seleccionados a un estado anterior.

• Control de versión usando una herramienta de compilación de cualquier Control de Código Fuente Común (SCC).

• Seguridad de usuario y grupo integrada, y administración del control de acceso.

• Soporte de pruebas para casos de prueba, JUnit y NUnit.

• Banco de Trabajo de Depuración integrado para perfilar aplicaciones Java y .Net ejecutables, instanciando objetos del modelo durante la ejecución y generando diagramas de Secuencia desde la traza de la pila.

• Opciones de documentación flexible y comprensiva: reportes escritos en MS Word compatibles con HTML y RTF.

• Soporte para ingeniería directa y reversa para muchos lenguajes “fuera de la caja”: ActionScript 3.0, Java, C#, C++, VB.Net, Delphi, Visual Basic , Python y PHP

• Desarrollo distribuido a través de archivos compartibles, usando repositorios compartidos en el rango de sistemas de administración de base de datos, replicación de archivos, transferencia de datos, e importación y exportación de datos de referencia.

• Habilidad para llevar a cabo el modelado de bases de datos, hacer ingeniería reversa desde un rango de DBMS, y generar directamente scripts DDL para crear estructuras de bases de datos.

• Habilidad de importar o exportar modelos a archivos XMI 2.1.

• Transformaciones (MDA) integradas, y facultades para importar o crear otras.

• Facultades para importar esquemas de base de datos, XSD y fuente WSDL, binarios Java y .Net.

• Kit para desarrolladores de software para scriptear y personalizar Enterprise Architect.

• La vista de solo-lectura permite a los accionistas ver pero no cambiar hitos deliberadamente.

• Un rango interno y externo de Add-Ins MDG comerciales para integrar las facultades de Enterprise Architect con IDEs y otras tecnologías, y plantillas para que diseñe la suya.

• Control de ortografía del Proyecto.

### 3.2.4 Herramienta NDT Suite

Navigational Development Techniques (NDT) es una metodología para el desarrollo de sistemas Web e hipermedia. NDT-Suite es un conjunto de herramientas para aplicar la metodología NDT.

NDT-Suite soporta las fases de requisitos, análisis, diseño, construcción e implantación, pruebas y mantenimiento, todas ellas basadas en la metodología Métrica V3. La fusión realizada se basa en definir un proceso, similar al de Métrica V3 pero haciendo uso de los modelos de UML y de las extensiones que NDT realiza de ellos, así como de sus procesos de ingeniería guiada por modelos.

NDT-Suite se compone de cuatro herramientas, que son: NDT-Profile, NDT-Quality y NDT-Driver.

# 4. RESULTADOS

## 4.1 Resultados obtenidos

### 4.1.1 Caso de uso

****

**4.1.2 Figura: Diagrama de Caso de uso**

### 4.1.3 Entidades y Relaciones

****

**4.1.4 Figura: Diagrama de Entidad Relación**

## 4.2 Resultados NDT

### 4.2.1 Participantes

****

**4.2.2 Figura: Diagrama de Participantes**

### 4.2.3 Actores

****

**4.2.4 Figura: Diagrama de Actores**

### 4.2.5 Objetivos

****

**4.2.6 Figura: Diagrama de Objetivos**

### 4.2.7 Navegación

****

**4.2.8 Figura: Diagrama de Navegación**

## 4.3 Prototipos

### 4.3.1 Página Principal

****

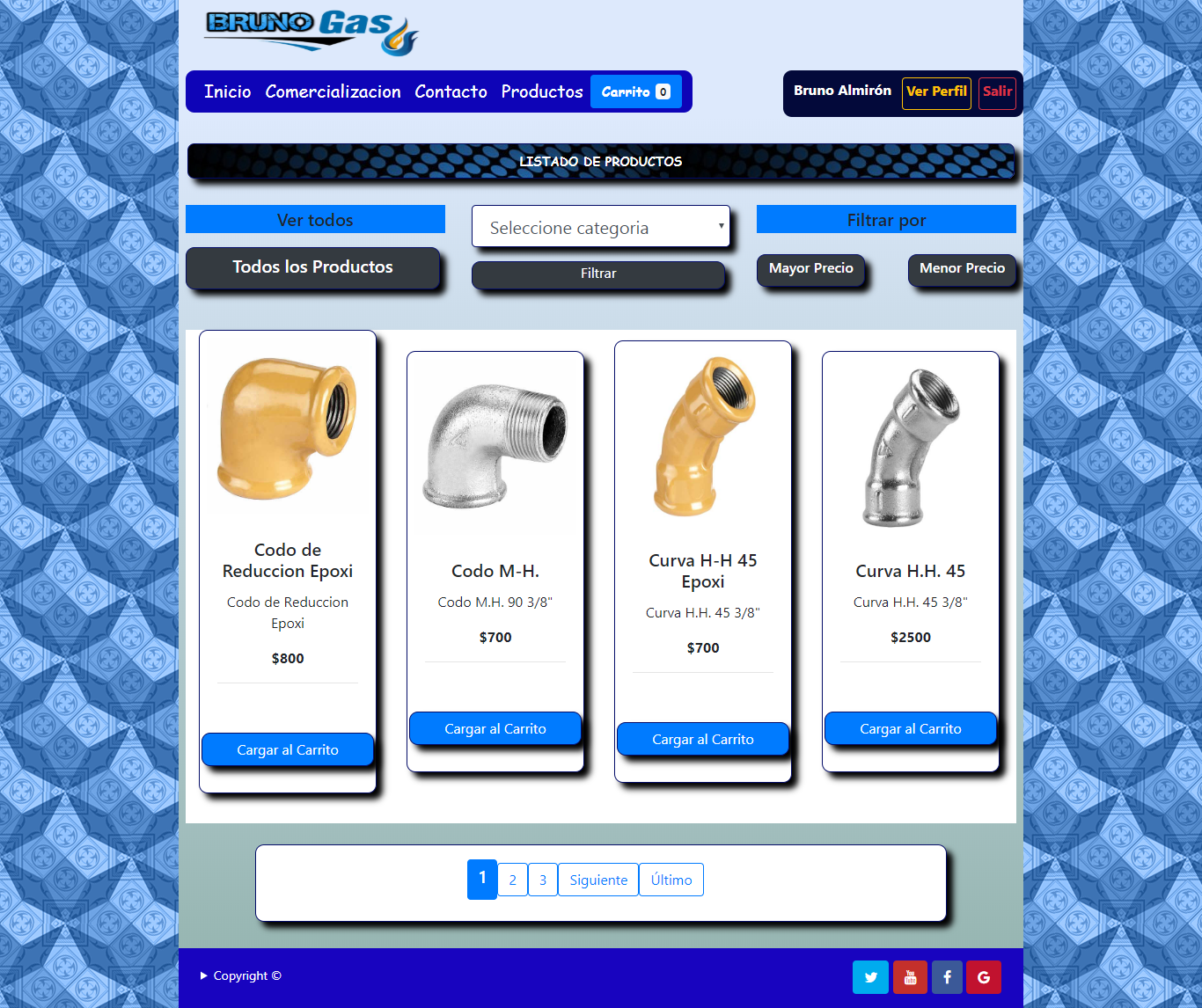
**4.3.2 Figura: Página principal del Sistema**

### 4.3.3 Login y Registro



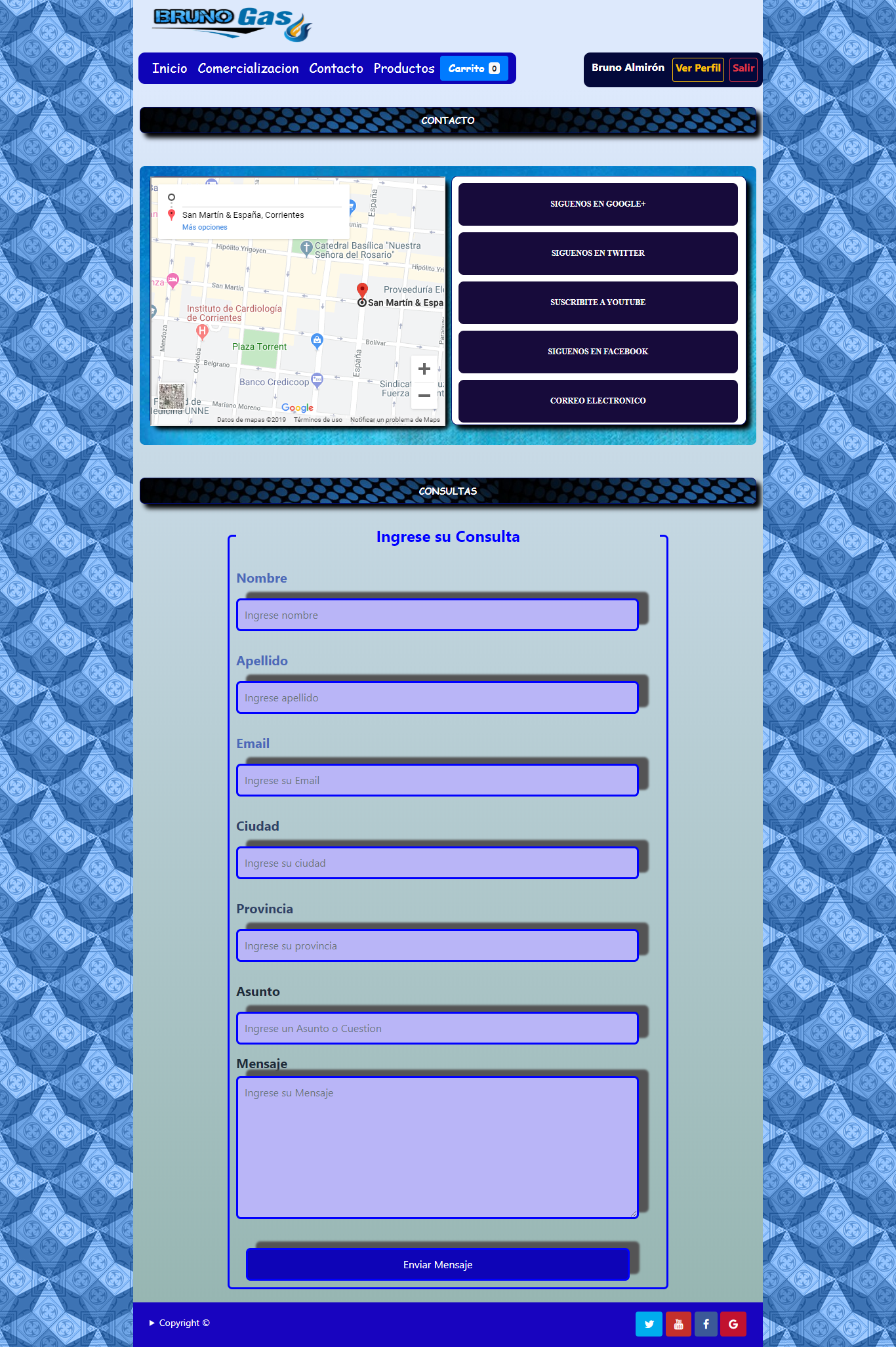
**4.3.4 Figura: Iniciar Sesión o Identificarse**

### 4.3.5 Catalogo

****

**4.3.6 Figura: Catalogo de Productos**

### 4.3.7 Consultas

****

**4.3.8 Figura: Consultas**

### 

# 5. CONCLUSIONES

La ingeniería de software proporciona técnicas y procedimientos que permiten ayudar a asegurar un software de calidad.

Cada vez más la ingeniería de software es necesaria en el desarrollo de sistemas embebidos de calidad debido a:

* Crecientes posibilidades del hardware
* Creciente complejidad y tamaño en el software
* Distribución y composición de equipos de trabajo

## 

## 5.1 Referencias

Reglamento del Trabajo Final de Carrera. LICENCIATURA EN SISTEMAS DE INFORMACION.

Anexo III: Documentación a presentar para la solicitud de defensa del PFC.

Vallejos, Oscar A. 2018. Plantillas de Teoría “Ingeniería del Software I”

Gómez Solís, Laura 2018. Plantillas de Práctica “Ingeniería del Software I”

Mattoso, Alejandra2018. Plantillas de Práctica “Ingeniería del Software I”

Pressman, Roger S. 2002. “INGENIERÍA DEL SOFTWARE. Un enfoque práctico. 5ta edición”. McGraw-Hill. Madrid. ISBN: 0-07709677-0.

Sommerville, Ian. 2005. “Ingeniería del software. Séptima edición”. Pearson Educación. Madrid. ISBN: 84-7829-074-5.

Yourdon, Edward. 1993. “Análisis Estructurado Moderno”. Prentice-Hall Hispanoamericana. Naucalpan de Juárez, Edo. de México. ISBN: 968-880-303-0.