**DOCUMENTACION DEL PROYECTO PARA LA MATERIA DE DISEÑO WEB III**

**PRESENTADO POR:**

**EDER BARRIOS CAMARGO**

**COD. (TOOO30535)**

**PROFESOR:**

**JAIRO ENRIQUE SERRANO CASTAÑEDA**

**UNIVERSIDAD TECNOLOGICA DE BOLIVAR**

**INGENIERIA DE SISTEMAS**

**CARTAGENA**

**2014**

**Introducción**

La documentación en un proyecto de software es crucial para comprender las relaciones, ideas, y lógica de la aplicación, por eso es tan importante al momento de desarrollar proyectos de software, documentar.

En el presente trabajo documenta un sistema para la gestión de la información de trapiches, un trapiche es el lugar donde se lleva a cabo el proceso de elaboración de la panela, en el cual entran usuarios, objetos que más adelante se detallaran.

El hecho de que un sistema este documentado, proyecta la seguridad de que cada una de las funcionalidades brindadas por este se han analizado, de tal forma que quede una constancia escrita y/o digital.

**Objetivos**

* Mostrar al cliente de manera clara las funcionalidades del sistema.
* Incentivar el desarrollo documentado.
* Retroalimentar temas sobre documentación.

**Sistema de información para la gestión del control administrativo**

**Del Trapiche XXX**

1. **RESULTADOS**

Para el desarrollo del sistema de información para la gestión del control administrativo del trapiche XXX utilizamos el modelo de construcción de software RUP (Rational Unified Process).

Primero se realizó un modelo de análisis donde se entendió de forma completa: cuál es el problema, haciendo énfasis en cada una de las necesidades para llevar a cabo el proyecto, incluyendo tanto requerimientos funcionales como no funcionales.

Luego se realizó un análisisy diseño enfocado en el dominio del problema y estableciendo una arquitectura de software sólida.

Seguido a esto, se hizo la Implementación de la aplicación, donde se plasmo todo lo referente al desarrollo de la aplicación, es decir, se transformó el diseño en código.

Y por último se realizaron diversas pruebas  al sistema con el fin de verificar  la funcionalidad en todos los aspectos de la aplicación desarrollada.

* 1. **MODELO DEL ANÁLISIS**
     1. **Análisis del sistema actual**
     2. **Modelo del mundo real.**

Por medio de este diagrama se modelan las clases del mundo real y las relaciones existentes entre las mismas, este permite visualizar los manejos de la información y analizar con mayor detenimiento cómo se coordinan los procesos en el trapiche xxx con respecto a la problemática de estudio de ésta investigación. A continuación el diagrama de dominio.

* + 1. **Requerimientos funcionales**
* La aplicación debe permitir el ingreso de múltiples usuarios, cada uno con nombre de usuario y contraseña que permita la confidencialidad de los datos y el control de acceso según los permisos asignados. **(3-4 semanas)**
* Tener un proceso que permita  agregar, modificar y/o eliminar datos de configuración, tales como: tipos de identificación, parámetros,  unidades de medida, tipos de operarios, tipos de herradura, etc.**(+2 semanas)**
* De igual forma, agregar, modificar y/o eliminar datos operativos tales como cosecheros, mulas, sembrados, operarios y arrieros. **(+2 semana)**
* Incluir un proceso que permita registrar una nueva molienda y realizar la inscripción de operarios y gestión de aprontes.**(+2 semanas)**
* Así mismo, registrar el cierre de molienda, liquidación a cosecheros y operarios y registrar los gastos de operación del trapiche.
* Y por último, debe con un módulo de reportes **(+2 semana)**
  + 1. **Requerimientos no funcionales.**

Se dividen en factores de calidad, requerimientos de hardware, requerimientos de software, requerimientos de programación, como se amplia a continuación:

* **Factores de Calidad**

**Escalabilidad:** Es escalable porque permite el crecimiento a nivel de usuarios. Se brinda acceso para múltiples usuarios identificando a cada uno con un *nombre de usuario* y una *contraseña*, garantizando privacidad e integridad con respecto a la información de los usuarios. Se debe poseer un mecanismo para la administración de los usuarios, es decir, darlos de alta, de baja o modificar los datos de los mismos. Teniendo como delimitación la capacidad del servidor para soportar accesos múltiples y simultáneos.

**Heterogeneidad:** La aplicación puede ser utilizada desde cualquier explorador como Internet Explorer, Mozilla, Chrome, entre otros, inherente a la plataforma en la que se ejecuta.

**Fiabilidad:** Se requiere de una aplicación que responda de manera oportuna y eficiente a cada una de las operaciones requeridas en un momento dado, para lo cual, se trató de forma minuciosa cada una de las operaciones o funciones brindadas por el sistema de información para que respondan con la precisión requerida por parte de los usuarios finales y en el menor tiempo de respuesta posible para de este modo garantizar que la aplicación es correcta en realizar cada una de las funcionalidades que se le soliciten en determinado momento. Esto se verificó al momento de realizar las pruebas piloto.

**Rendimiento:** se manejó la aplicación de manera que los tiempos de respuesta sean mínimos y satisfactorios para el usuario final y de este modo brindar un manejo eficiente del sistema. Este aspecto se tuvo en cuenta durante todo el período de prueba y se corroboró que efectivamente se están consiguiendo tiempos óptimos en las operaciones realizadas por el sistema.

**Implementación:** Para el desarrollo de la aplicación se utilizaron herramientas muy potentes y apropiadas para aplicaciones de este tipo, como son el lenguaje de programación PHP (Laravel)  y el motor de base de datos SQL Server.

**Facilidad de uso:** Se requiere que la aplicación sea de fácil manejo para todos los usuarios finales, incluso, si no han tenido antes contacto con una aplicación de este tipo. Se distribuyó y se adecuo dentro de cada interfaz los datos de manera que los usuarios tengan alcance inmediato y sencillo a toda la información manejada en ese momento. Se usarán menús sencillos de acceso a las interfaces relacionadas con la funcionalidad que se esté manejando, así como de acceso a toda la aplicación, pero de forma sencilla, para no recargar al usuario con tantos enlaces, de este modo se busca garantizar la facilidad de acceso, ítem importante en el desarrollo de una aplicación.

* **Requerimientos de hardware:**
* **Requerimientos de programación:**

Los requerimientos de programación a utilizar en el desarrollo de este proyecto son:

* 1. **ANÁLISIS Y DISEÑO**
     1. **Casos de uso del análisis.**

Para el análisis del sistema, se realizaron los casos de uso del análisis,  los cuales permitieron conocer el estado del sistema actual y las funciones de cada actor, al describir cada uno de los casos de uso del análisis se ven reflejados los requerimientos funcionales para el desarrollo de un buen diseño del sistema de información. Estos casos de uso no se incluyeron en este documento ya que no son  programables y solo nos ayudan a entender la problemática estudio de este proyecto investigativo.

* + 1. **Glosario de Términos.**

A continuación se presentan los términos manejados a lo largo de todo el proyecto:

* Trapiche
* Apronte
* Operario
* Cosechero
* Arriero

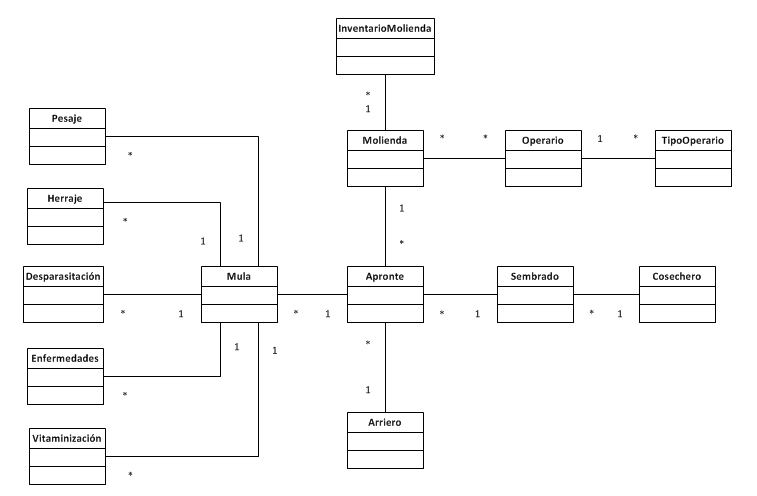
* 1. **DISEÑO DEL SISTEMA**

Para la elaboración del diseño del sistema, se procedió a realizar diagramas utilizando el lenguaje de modelado unificado (UML).

En primer lugar se presenta el diagrama de clases el cual, se fundamenta en el diagrama de dominio, ya que este proporciona los elementos que se identificaron en el modelo del mundo real. Utilizando el lenguaje de modelado unificado se trazaron las relaciones y cardinalidades entre las clases.

El diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Describiendo el diagrama de clases que se muestra a continuación se tiene que una **Molienda** se inscribe uno o más **operarios...**



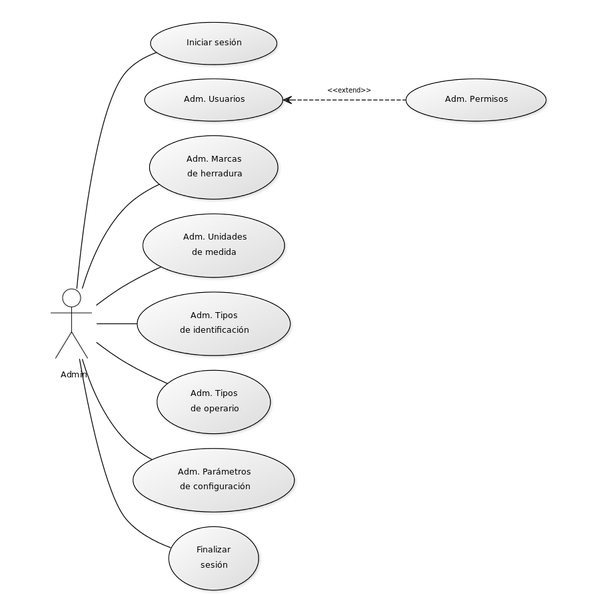
* + 1. **Casos de uso del sistema**

Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema realiza y que produce un resultado observable de un valor o actor particular. El caso de uso se utiliza para estructurar el comportamiento de los elementos en un modelo.

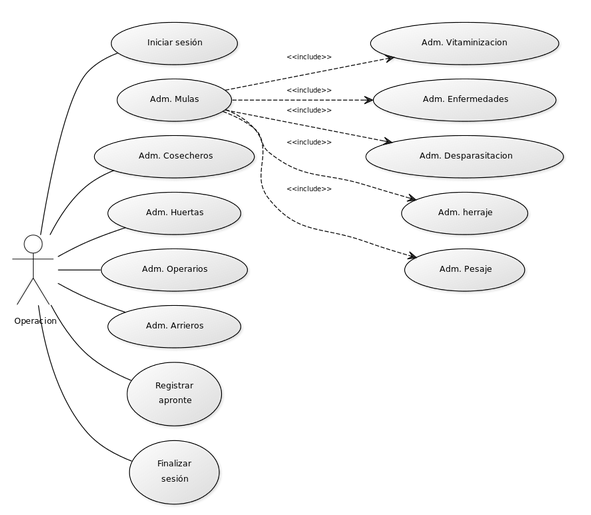
* + - 1. **Descripción de Actores del Sistema**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Casos de uso** | **Tipo** |
| Administrador | Iniciar sesión, administrar usuarios, permisos, tipos de identificación, unidades de medida, marcas de herraduras, tipos de operario, parámetros de configuración | Primario |
| Operación | Iniciar sesión, administrar cosecheros, mulas, huertas, operarios, arrieros y aprontes. | Primario |
| Financiero | Iniciar sesión,nueva molienda, cierre de molienda, liquidación cosecheros, liquidación operarios, gastos de molienda y reportes. | Primario |

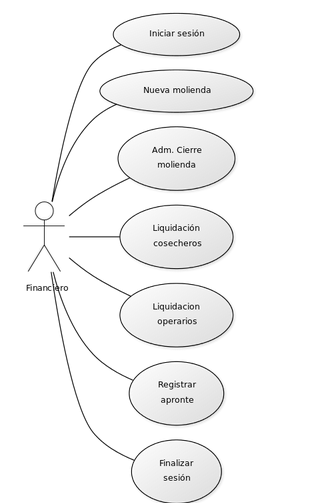
* + - 1. **Diagramas de Casos de Uso** 
         * **Administrador**

****

* + - * + **Operación**

****

* + - * + **Financiero**

****