**Facultad de Ingeniería**

**Ingeniería Civil Informática**



**DESARROLLO DE UNA APLICACIÓN WEB PARA EL MONITOREO DE FLOTA DE MAQUINARIA PESADA QUE PRESTA SERVICIO A FORESTAL ARAUCO**

**PATRICIO ANDRÉS VILLANUEVA FUENTES**

Informe de proyecto de título para optar al título de:

**INGENIERO CIVIL INFORMÁTICO**

Profesor Guía:

Manuel Novoa Olivares

Concepción, 30 de Junio de 2017

RESUMEN

ABSTRACT

AGRADECIMIENTOS

ÍDICE DE CONTENIDOS

[1 CAPÍTULO 1: INTRODUCCIÓN 1](#_Toc485149334)

[1.1 PRESENTACIÓN DEL TEMA 1](#_Toc485149335)

[1.2 OBJETIVOS GENERALES 1](#_Toc485149336)

[1.3 OBJETIVOS ESPECÍFICOS 1](#_Toc485149337)

[1.4 JUSTIFICACIÓN DEL PROBLEMA 1](#_Toc485149338)

[1.5 DELIMITACIÓN DEL PROBLEMA 2](#_Toc485149339)

[1.6 METODOLOGÍA APLICADA 2](#_Toc485149340)

[1.7 CAÍTULOS SIGUIENTES 3](#_Toc485149341)

[2 CAPÍTULO 2: MARCO TEÓRICO Y SITUACIÓN ACTUAL DE LA EMPRESA 3](#_Toc485149342)

[2.1 MARCO TEÓRICO 3](#_Toc485149343)

[2.1.1 LA EMPRESA 3](#_Toc485149344)

[2.1.2 MAQUINARIA PESADA 3](#_Toc485149345)

[2.1.3 DISTANCIA ENTRE DOS PUNTOS GEOGRÁFICOS 4](#_Toc485149346)

[2.1.4 INGENIERÍA Y PROCESO DEL SOFTWARE 4](#_Toc485149347)

[2.1.5 HTML 5 5](#_Toc485149348)

[2.1.6 LIBRERÍAS Y FRAMEWORKS 6](#_Toc485149349)

[2.1.7 MYSQL Y PHP 6](#_Toc485149350)

[2.1.8 LICENCIAS DE CÓDIGO 6](#_Toc485149351)

[2.1.9 DISEÑO ADAPTABLE 6](#_Toc485149352)

[2.1.10 INTERFAZ Y EXPERIENCIA DE USUARIO 6](#_Toc485149353)

[2.2 SITUACIÓN ACTUAL DE LA EMPRESA 6](#_Toc485149354)

[3 CAPÍTULO 3: ESPECIFICACIÓN DE REQUERIMIENTOS 7](#_Toc485149355)

[3.1 INTRODUCCIÓN 7](#_Toc485149356)

[3.1.1 PROPÓSITO 7](#_Toc485149357)

[3.1.2 ÁMBITO DEL SISTEMA 7](#_Toc485149358)

[3.1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS 8](#_Toc485149359)

[3.1.4 REFERENCIAS 8](#_Toc485149360)

[3.1.5 VISIÓN GENERAL DEL DOCUMENTO 8](#_Toc485149361)

[3.2 DESCRIPCIÓN GENERAL 8](#_Toc485149362)

[3.2.1 PERSPECTIVA DEL PRODUCTO 8](#_Toc485149363)

[3.2.2 FUNCIONES DEL PRODUCTO 8](#_Toc485149364)

[3.2.3 CARACTERÍSTICAS DE LOS USUARIOS 8](#_Toc485149365)

[3.2.4 RESTRICCIONES 8](#_Toc485149366)

[3.2.5 SUPOSICIONES Y DEPENDENCIAS 8](#_Toc485149367)

[3.3 REQUISITOS ESPECÍFICOS 8](#_Toc485149368)

[3.3.1 INTERFACES EXTERNAS 8](#_Toc485149369)

[3.3.2 FUNCIONES 8](#_Toc485149370)

[3.3.3 REQUISITOS DE RENDIMIENTO 8](#_Toc485149371)

[3.3.4 RESTRICCIONES DE DISEÑO 8](#_Toc485149372)

[3.3.5 ATRIBUTOS DEL SISTEMA 8](#_Toc485149373)

[3.3.6 OTROS REQUISITOS 8](#_Toc485149374)

[4 CONTRUCCIÓN Y PRUEBAS 8](#_Toc485149375)

[5 CONCLUSIONES Y RECOMENDACIONES 8](#_Toc485149376)

[6 BIBLIOGFÍA 8](#_Toc485149377)

[7 ANEXOS 11](#_Toc485149378)

ÍNDICE DE TABLAS

ÍNDICE DE ESQUEMAS

ÍNDICE DE FIGURAS

NOMENCLATURA Y ABREVIACIONES

# CAPÍTULO 1: INTRODUCCIÓN

## PRESENTACIÓN DEL TEMA

Forestal Arauco está desarrollando dispositivos de seguimiento para la maquinaria de extracción de árboles en las zonas aledañas de la región del Bío-Bío, es por esto que es necesario un sistema web para el monitoreo y seguimiento de los sensores que se instalarán en cada máquina.

El monitoreo de los dispositivos a instalar en cada máquina, tiene como objetivo saber si la máquina está realmente operando o cumpliendo su función, para esto, se tiene considerado medir las siguientes variables, pudiendo adicionarse algunas más con el pasar del tiempo: velocidad de desplazamiento, posición de la máquina, ángulo de inclinación de la pala, revoluciones del motor y ángulo de inclinación de la máquina.

Actualmente existen 60 zonas de trabajo en las cuales operan entre 10 y 15 máquinas.

La aplicación debe ser adaptable a dispositivos móviles y contar con un sistema de geo localización de dispositivos.

## OBJETIVOS GENERALES

Desarrollar una aplicación web responsiva que permita monitorear las características de las máquinas en operación que prestan servicio a forestal Arauco.

## OBJETIVOS ESPECÍFICOS

* Definir los lenguajes, bibliotecas y frameworks a usar para el desarrollo de la aplicación.
* Construir múltiples prototipos según los requerimientos del cliente.
* Determinar el tiempo de inactividad diario de las máquinas operativas.
* Generar datos que ayuden a la toma de decisiones de la empresa.
* Crear gráficos para la representación de los datos.

## JUSTIFICACIÓN DEL PROBLEMA

* Se ha detectado que las máquinas contratadas pasan más del 25% del tiempo estacionadas o no operando durante el horario de trabajo.
* La empresa pierde recursos por un servicio costoso y que debería estar 100% operativo durante el tiempo que se les contrata.
* Se necesita supervisar cada máquina, por lo que contratar a un supervisor sería aún más costoso.

## DELIMITACIÓN DEL PROBLEMA

* Los lenguajes de programación a utilizar son: PHP y JavaScript.
* Los problemas de señal de los dispositivos y cobertura del mismo, no son parte del proyecto.
* En una primera fase, se harán pruebas con 10 máquinas, para luego abarcar la totalidad de las zonas aledañas de la región.
* La plataforma web funcionará en navegadores que soporten la API de Geolocalización de google.

## METODOLOGÍA APLICADA

Una metodología de desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado, lo que nos permitirá una forma sistemática para poder obtener un producto correcto y libre de errores.

Para este proyecto se hará uso de la metodología por prototipos tomando en cuenta los siguientes antecedentes:

* El cliente no conoce cuales son las especificaciones de forma precisa
* El proyecto cuenta con desarrollos de productos con innovaciones importantes.
* Ayuda al desarrollador y a otros participantes a mejorar la compresión de lo que hay que elaborar cuando los requerimientos no están claros.
* El cliente necesita con urgencia la aplicación web.

El modelo de prototipos comienza con la comunicación entre clientes y desarrolladores, en donde se definen los objetivos generales del software, se identifican los requerimientos que se conocen y se detectan las áreas en las que es fundamental una mayor definición. Luego se planea rápidamente una iteración para realizar el prototipo, para luego pasar a la etapa del modelado del mismo.

El diseño rápido lleva a la construcción de un prototipo, el cual se entrega y es evaluado y validado por los participantes, los cuales dan retroalimentación para mejorar o decidir nuevos requerimientos para el sistema.

Finalmente, luego de varias iteraciones, el prototipo se transforma en el sistema real (prototipo evolutivo).

## CAÍTULOS SIGUIENTES

# CAPÍTULO 2: MARCO TEÓRICO Y SITUACIÓN ACTUAL DE LA EMPRESA

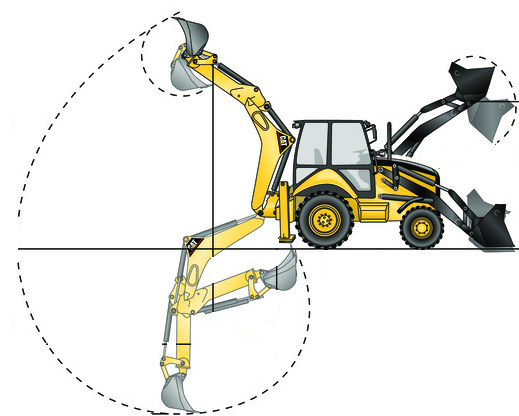
## MARCO TEÓRICO

### LA EMPRESA

### MAQUINARIA PESADA

Las maquinarias pesadas se definen como vehículos destinados exclusivamente a obras industriales incluidas las de minería, construcción y conservación de obras. Se pueden clasificar según las operaciones comunes que realizan: excavar, transportar, cargar, conformar, humedecer y compactar.

Una de las máquinas más utilizadas en la industria forestal es la retroexcavadora mixta, ésta máquina dispone de una pala ancha capaz de mover volúmenes considerables de tierras y por otro lado dispone de una pala con brazo articulado muy práctica para la ejecución de zanjas, trabajos en taludes, movimiento de escombros etc. Por su reducido volumen es capaz de moverse en terrenos difíciles y es imprescindible para toda empresa dedicada al movimiento de tierras y/o construcción.



### DISTANCIA ENTRE DOS PUNTOS GEOGRÁFICOS

Una de las fórmulas más utilizadas y sencillas para el cálculo de la distancia entre dos puntos es la fórmula de Haversine, esta fórmula se basa en el cálculo de la distancia de círculo máximo entre dos puntos de un globo a partir de su longitud y latitud, la fórmula consisten en:

### INGENIERÍA Y PROCESO DEL SOFTWARE

La ingeniería de software es una especialidad de la ingeniería que abarca todos los temas de la producción de software desde el inicio de las especificaciones del sistema, hasta el mantenimiento del mismo después que se utiliza.

“El proceso de software es un conjunto de actividades que conducen a la creación de un producto software” (Pressman).

Existen diversos procesos diferentes de software, algunas actividades comunes entre los procesos son:

* Especificación del software: etapa donde se definen las funcionalidades del software y las restricciones en su operación, comúnmente se realiza una especificación de requerimientos de software (ERS), documento que se crea para especificar detalladamente todos los aspectos del software antes de que el proyecto comience. La ERS se justifica cuando el software va a ser desarrollado por una tercera persona, cuando la falta de una especificación crearía problemas graves al negocio, si un sistema es muy complejo o si se trata de un negocio de importancia crítica.
* Diseño e implementación: el diseño comienza una vez se han analizado y modelado los requerimientos, esta etapa prepara la etapa de construcción o implementación del software en donde se genera y prueba código.
* Validación de software: etapa para mostrar que el sistema se ajusta a la especificación y que cumple con las expectativas del cliente.
* Evolución del software: etapa avanzada del software donde se modifica para la adaptación a los cambios requeridos por el cliente y el mercado.

### HTML 5

HTML5 es una especificación que combina HTML, CSS y JavaScript, tecnologías requeridas para construir la web que hoy disfrutamos, a continuación se describen cada una de estas tecnologías:

HTML es un lenguaje de compuesto por un grupo de etiquetas definidas con un nombre rodeado de unos paréntesis angulares. Los paréntesis angulares delimitan la etiqueta y el nombre define el tipo de contenido que representa. Alguna de estas etiquetas son declaradas individualmente y otras son declaradas en pares, con una etiqueta de apertura y otra de cierre. Las etiquetas individuales y las de apertura pueden incluir atributos para ofrecer información adicional acerca de sus contenidos, etiquetas individuales y la combinación de etiquetas de apertura y cierre son llamados elementos. Los elementos compuestos por una sola etiqueta son usados para modificar el contenido que los rodea o incluir recursos externos, mientras que los elementos que incluyen etiquetas de apertura y cierre son utilizados para delimitar el contenido del documento.

CSS es el lenguaje utilizado para definir los estilos de los elementos HTML, como el tamaño, el color, el fondo, el borde, etc. Para declarar estilos personalizados, CSS utiliza propiedades y valores, esta construcción es llamada declaración y su sintaxis incluye dos puntos, luego el nombre de la propiedad y un punto y coma al final para cerrar la línea.

JavaScript, a diferencia de HTML y CSS es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, es decir, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Actualmente todos los navegadores modernos incluyen un intérprete o motor de JavaScript, que traduce e interpreta las instrucciones escritas en este lenguaje.

En conclusión, HTML se encarga de definir la estructura del documento, CSS prepara esa estructura y su contenido para ser mostrado en pantalla, y JavaScript introduce la capacidad de procesamiento necesaria para construir aplicaciones web completamente funcionales.

### LIBRERÍAS Y FRAMEWORKS

Una librería se define como un conjunto de código que contiene funciones que permiten realizar tareas que nos facilitan la programación

### MYSQL Y PHP

MySQL es un SABD relacional rápido y fácil de utilizar, alguna de sus características son:

* Facilidad de uso, se puede construir una base de datos MySQL e interactuar con ella usando unas cuantas instrucciones simples en el lenguaje SQL.
* Compatibilidad con sistemas operativos, funciona con una gran variedad de sistemas operativos como: Windows, Linux, Mac OS, FreeBSD, OS/2, Irix, Solaris, AIX DEC Unix, etc.
* Soporte de base de datos grandes, maneja base de datos de hasta 50 millones de filas o más. El límite predefinido para el tamaño de archivo de una tabla es de 4GB, pero se puede aumentar hasta un límite de 8 millones de terabytes.

PHP es un potente lenguaje de programación cuya facilidad de uso tiene mucho que ver con la posibilidad de incrustar pequeños fragmentos de códigos en el documento HTML, alguno de los beneficios de usar PHP son:

* Seguridad, el código generado en PHP no se puede ver desde el navegador.
* Soporte, dispone de una abundante comunidad web al ser uno de los lenguajes más populares.
* Lenguaje de código abierto y la curva de aprendizaje es muy baja.
* Facilidad de acceso a base de datos.

### LICENCIAS DE CÓDIGO

### DISEÑO ADAPTABLE

### INTERFAZ Y EXPERIENCIA DE USUARIO

## SITUACIÓN ACTUAL DE LA EMPRESA

# CAPÍTULO 3: ESPECIFICACIÓN DE REQUERIMIENTOS

## INTRODUCCIÓN

### PROPÓSITO

El objetivo de este documento es definir de manera clara y precisa las funcionalidades y restricciones que tendrá el sistema que se desea construir.

Este documento va dirigido a la empresa ARAUCO SA y a los desarrolladores del sistema a implementar.

### ÁMBITO DEL SISTEMA

‘Machine Monitors’, nombre que se le dará al sistema, tiene como objetivo monitorear el comportamiento de la maquinaria pesada que presta servicio a forestal ARAUCO SA, esperando así que se facilite la interpretación y lectura de información de la variables medidas por los sensores para la ayuda de toma de decisiones de la empresa.

Los beneficios que se esperan obtener con el sistema son:

* Seleccionar maquinas cuyo rendimiento no sea el esperado.
* Controlar si las máquinas están realmente habilitadas o en funcionamiento.
* Reducir el gasto en la contratación del servicio de arriendo de maquinaria pesada.

### DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

LEER LA WEA DENUEVO Y SACAR TERMINOS

### REFERENCIAS

PONER EL PRESSMAN Y IEEE 380

### VISIÓN GENERAL DEL DOCUMENTO

## DESCRIPCIÓN GENERAL

### PERSPECTIVA DEL PRODUCTO

PONER LO DEL CELU Y LAS INTERFACES BOTONES MENUES ETC Y

### FUNCIONES DEL PRODUCTO

PONER LAS FUNCIONES A CAGAR NOMA CON FOTOS

### CARACTERÍSTICAS DE LOS USUARIOS

PONER LA TABLA CULIA

### RESTRICCIONES

#### LIMITACIONES DE HARDWARE

##### PARA EL DESARROLLO

Se consideraran las siguientes características físicas mínimas del computador:

* + CPU de 2.66 GHZ
  + 2GB de RAM
  + 80 GB de Disco Duro
  + Tarjeta de video integrada
  + Mouse o touchpad
  + Teclado
  + Pantalla

##### PARA LA USABILIDAD

Se considerarán las siguientes características físicas mínimas del computador:

* + CPU de 2 GHZ
  + 512 de RAM
  + Tarjeta de video integrada
  + Mouse o touchpad
  + Teclado
  + Pantalla

Se consideraran las siguientes características físicas mínimas para dispositivos móviles

* + CPU de 1 GHZ
  + 512 MB de RAM

#### INTERFACES CON OTRAS APLICACIONES

El sistema es independiente, no interactúa con ningún otro sistema de la empresa.

El sistema realizara conexiones con una base de datos MySQL a través de los navegadores web y los lenguajes de programación PHP y JavaScript.

#### LENGUAJES DE PROGRAMACION

El sistema estará basado en HTML5, PHP y SQL.

#### PROTOCOLOS DE COMUNICACIÓN

El sistema hará uso del protocolo TCP/IP y HTTP.

#### CONSIDERACIONES ACERCA DE LA SEGURIDAD

Considerando que el sistema está en un ambiente web se utilizará:

* + Método POST para envió de información.
  + Método GET combinado con funciones hash MD5 o SHA1.
  + Variables de sesión.
  + Algoritmo Blowfish para contraseñas con hash (recomendado por PHP).
  + Funciones de PHP para prevenir inyecciones SQL.

### SUPOSICIONES.

El sistema se alimenta de los datos de los archivos, es por esto que el sistema supone que los supervisores subirán los datos de los archivos frecuentemente.

## REQUISITOS ESPECÍFICOS

### REQUISITOS DE DESARROLLO

* + Entorno web local: XAMPP.
  + Sistema gestor de base de datos: MySQL.
  + Office Visio 2013: Diagrama de casos de usos y diagramas de actividad.
  + Navegadores web (últimas versiones): Firefox, Google Chrome y Safari.
  + Lenguajes de programación: JavasScript y PHP.
  + Framework CSS: Bootstrap.
  + Control de versiones: GIT.

### FUNCIONES

COLOCAR LOS REQUISITOS

### REQUISITOS DE RENDIMIENTO

PONER QUE SE ESTIMA QUE SE SUBAN MUCHOS DATOS Y QUE EL SERVIDOR ALOJADO DEBE TENER LA CAPACIDAD DE AUMENTAR TAMANO BUSCAR CUANTO SOPORTA MYSQL

### ATRIBUTOS DEL SISTEMA

BUSCAR NO FUNCIONALES DE CALIDAD

# CONTRUCCIÓN Y PRUEBAS

# CONCLUSIONES Y RECOMENDACIONES

# BIOGRAFÍA

* Aubry, C. (2014). HTML5 y CSS3 para sitios web con Diseño Web Responsive. Barcelona, España: ENI.
* Aubry, C. (2015). CC3 Domine los estándares web con las hojas de estilo. Barcelona, España: ENI.
* Bootstrap Foundation. (2016).  Grid System. Recuperado de:  http://getbootstrap.com/css/#grid
* Chaffer, J. (2009). Aprende jQuery 1.3.  Madrid, España: ANAYA.
* De Luca, Damian. (2011). HTML5. Buenos Aires: Argentina: Redusers.
* Google Developers. (2016). API de Google Maps. Recuperado de: https://developers.google.com/maps/documentation/javascript/?hl=es
* Google Developers. (2016). Google Charts. Recuperado de: https://developers.google.com/chart/
* Google Developers. (2016).  Using Google Chart. Recuperado de: <https://developers.google.com/chart/interactive/docs/>
* Heurtel O. (2014). Php 5.5 Desarrollar un sitio web dinámico e interactivo. Barcelona, España: ENI.
* Jquery Foundation. (2016). Jquery API. Recuperado de: http://api.jquery.com/
* Magno, A. (2013). Mobile First Bootstrap. Birminham, UK: Packt Publishing Ltd.
* Marín, J. (2010). Web 2.0, Una descripción muy sencilla de los cambios que estamos viviendo, La Coruña, España: Netbiblo.
* Matarazzo, D. (2015). Aprenda los los lenguajes HTML5, CSS3 y JavasScript para crear si primer sitio web, Barcelona, España: ENI.
* Monteiro, F. (2014). Learning Single-page Web Application Development. Birminham, UK: Packet Publishing Ltd.
* Mozilla developer. (2016). Ajax. Recuperado de: <https://developer.mozilla.org/en-US/docs/AJAX>.
* Mozilla developer. (2016). XMLHttpRequest. Recuperado de: https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest
* Nafría, I. (2008). Web 2,0. Barcelona, España: Gestión 2000.
* Lockhart, J. (2015). Modern Php. New York, EEUU: O'reilly.
* Pressman, R. (2002). Ingeniería de software, un enfoque práctico Madrid, España: The McGraw Hill Companies.
* Rollet, O. (2015). Aprender a desarrollar un sitio web con php y mysql, Barcelona, España: ENI.
* Ruthkoski, T. (2013). Google Visualization API Essentials. Birminham, UK: Packet Publishing Ltd.
* Sommerville I. (2005). Ingeniería del software. Madrid, España: Pearson Educación S.A.
* Spurlock, J. (2013). Bootstrap. New York, EEUU: O'reilly.
* Svennerberg, G. (2010). Beginning Google Maps API 3. New York, EEUU: Apress.
* Van Lancker, L. (2013). Los API Javascript de HTML5. Barcelona, España: ENI.
* W3C. (2016). Estándares. Recuperado de:  http://www.w3c.es/estandares/
* W3School, (2016). Ajax Tutorial. Recuperado de: http://www.w3schools.com/ajax/

# ANEXOS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLIENTES** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idcliente** | identificador único y autoincremental del cliente | int(11) | si | no nulo |  |
| **correo** | correo del cliente | varchar(60) |  |  |  |
| **password** | contraseña del cliente | varchar(12) |  |  |  |
| **empresa** | empresa a la cual pertenece el cliente | varchar(45) |  |  |  |
| **cargo** | cargo del cliente | varchar(45) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLIENTES\_EMPRESAS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **id** | identificador único y autoincremental de la relación entre las tablas clientes y empresas | int(11) | si | no nulo |  |
| **idCliente** | id del cliente, referencia a la tabla clientes | int(11) |  |  | FK: tabla clientes |
| **idEmpresa** | id de la empresa, referencia a la tabla empresas | int(11) |  |  | FK: tabla empresas |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPRESAS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idEmpresa** | identificador único y autoincremental de la empresa | int(11) | si | no nulo |  |
| **rut** | rut de la empresa | varchar(10) |  |  |  |
| **correo** | correo de contacto de la empresa | varchar(60) |  |  |  |
| **nombre** | nombre de la empresa | varchar(45) |  |  |  |
| **telefono** | celular de contacto de la empresa | int(9) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ZONAS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idZona** | identificador único y autoincremental de la zona | int(11) | si | no nulo |  |
| **idEmpresa** | Id de la empresa, referencia a la tabla empresas | int(11) |  |  | FK: tabla empresas |
| **nombre** | Nombre de la zona | varchar(45) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MÁQUINAS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idMaquina** | identificador único y autoincremental de la máquina | int(11) | si | no nulo |  |
| **idZona** | id de la zona, referencia a la tabla zonas | int(11) |  |  | FK: tabla zonas |
| **patente** | patente de la máquina | varchar(6) |  |  |  |
| **fechaRegistro** | fecha en la cual se registró la máquina | date |  |  |  |
| **tara** | tara de la máquina en kilogramos | int(11) |  |  |  |
| **cargaMaxima** | carga máxima que soporta la máquina en kilogramos | int(11) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SUPERVISORES\_ZONAS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **id** | id de la relación entre las tablas supervisores y zonas | int(11) | si | no nulo |  |
| **idZona** | id de la zona, referencia a la tabla zonas | int(11) |  |  | FK: tabla zonas |
| **idSupervisor** | Id del supervisor, referencia a la tabla supervisores | int(11) |  |  | FK: tabla supervisores |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SUPERVISORES** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idSupervisor** | id único y autoincremental del supervisor | int(11) | si | no nulo |  |
| **nombreSupervisor** | nombre del supervisor | varchar(45) |  |  |  |
| **correoSupervisor** | correo del supervisor | varchar(60) |  |  |  |
| **password** | contraseña del supervisor | varchar(12) |  |  |  |
| **celular** | celular de contacto del supervisor | int(9) |  |  |  |
| **status** | indica el estado de la cuenta del supervisor, puede ser ‘habilitado’ o ‘deshabilitado’ | enum(‘habilitado’, ‘deshabilitado’) |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ARCHIVOS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idArchivo** | id único y autoincremental del archivo | int(11) | si | no nulo |  |
| **idZona** | Id de la zona, referencia a la tabla zonas | int(11) |  |  | FK: tabla zonas |
| **idSupervisor** | Id del supervisor, referencia a la tabla supervisores | int(11) |  |  | FK: tabla supervisores |
| **fechaSubida** | Fecha de subida de datos contenidos en el archivo | date |  |  |  |
| **fechaDatos** | Fecha a la cual corresponden los datos contenidos en el archivo | date |  |  |  |
| **horaSubida** | hora de subida de datos contenidos en el archivo | time |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DATOS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idDato** | id único y autoincremental del dato | int(11) | si | no nulo |  |
| **idArchivo** | Id del archivo al cual pertenece el resultado, referencia a la tabla archivos | int(11) |  |  | FK: tabla archivos |
| **patente** | Patente de la maquina | varchar(6) |  |  |  |
| **hora** | Hora del dato tomado | time |  |  |  |
| **latitud** | Latitud de la máquina | Float(10,6) |  |  |  |
| **Longitud** | Longitud de la máquina | Float(10,6) |  |  |  |
| **motorFuncionando** | Valor 1 si el motor estuvo funcionando o 0 si no estuvo funcionando | Int(1) |  |  |  |
| **Rpm** | Revoluciones por minuto de la máquina | float |  |  |  |
| **gradosPalaFrontal** | Grados pala frontal de la máquina | float |  |  |  |
| **gradosPalaTrasera** | Grados pala trasera de la máquina | float |  |  |  |
| **cambio** | Número de la marcha de la máquina | Int(2) |  |  |  |
| **alturaPalaFrontal** | Altura de la pala frontal de la máquina | float |  |  |  |
| **alturaPalaTrasera** | Altura de la pala trasera de la máquina | float |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RESULTADOS** | | | | | |
| **ATRIBUTO** | **DESCRIPCIÓN** | **TIPO** | **PK** | **NULL** | **FK** |
| **idDato** | id único y autoincremental del dato | int(11) | si | no nulo |  |
| **idArchivo** | Id del archivo al cual pertenece el resultado, referencia a la tabla archivos | int(11) |  |  | FK: tabla archivos |
| **patente** | Patente de la maquina | varchar(6) |  |  |  |
| **hora** | Hora del dato tomado | time |  |  |  |
| **latitud** | Latitud de la máquina | Float(10,6) |  |  |  |
| **Longitud** | Longitud de la máquina | Float(10,6) |  |  |  |
| **motorFuncionando** | Valor 1 si el motor estuvo funcionando o 0 si no estuvo funcionando | Int(1) |  |  |  |
| **Rpm** | Revoluciones por minuto de la máquina | float |  |  |  |
| **gradosPalaFrontal** | Grados pala frontal de la máquina | float |  |  |  |
| **gradosPalaTrasera** | Grados pala trasera de la máquina | float |  |  |  |
| **cambio** | Número de la marcha de la máquina | Int(2) |  |  |  |
| **alturaPalaFrontal** | Altura de la pala frontal de la máquina | float |  |  |  |
| **alturaPalaTrasera** | Altura de la pala trasera de la máquina | float |  |  |  |