

CAPITULO I - INTRODUCCIÓN .....	1
1.1 Antecedentes .....	1
1.2 Análisis del Problema .....	1
1.2.1 Definición del Problema .....	1
1.3 Objetivo del Proyecto .....	2
1.3.1 Objetivo General .....	2
1.3.2 Objetivos Específicos .....	2
1.4 Áreas Involucradas .....	2
1.5 Justificación .....	2
1.6 Alcance .....	3
CAPITULO II - MARCO TEORICO .....	4
2.1 Sistema Adminstrativo .....	4
2.1.1 Definicion .....	4
2.1.2 Sistema web .....	4
2.1.3 Cloud Computing .....	4
2.2 Metodologia del Proyecto .....	4
2.2.1 Metodologia Scrum .....	4
2.2.1.1 Definicion .....	4
2.2.1.2 Caracteristicas .....	5
2.2.1.3 Justificacion .....	5
2.3 Plataforma de desarrollo .....	5
2.3.1 Lenguaje de Programación Javascript.....	5
2.3.1.1 Definición .....	5
2.3.1.2 Caracteristicas .....	6
2.3.1.3 Justificacion .....	6
2.3.2 Framework Meteor .....	6
2.3.2.1 Definicion .....	6
2.3.2.2 Características .....	7
2.3.2.3 Justificación .....	7
2.3.3 Framework Semantic UI .....	8
2.3.3.1 Definicion .....	8
2.3.3.2 Caracteristicas .....	8
2.3.3.3 Justificación .....	8

2.4 Administrador de Base de Datos .....	8
2.4.1 MongoDB.....	8
2.4.1.1 Definición .....	8
2.4.1.2 Terminología básica de MongoDB .....	9
2.4.1.3 Arquitectura de MongoDB.....	9
2.4.1.4 Características .....	9
2.4.1.5 Justificación .....	10
CAPITULO III – Areas de aplicacion.....	11
3.1 Instalación deportiva .....	11
3.1.1 Teminologia.....	11
3.1.1.1 Espacio deportivo .....	11
3.1.1.2 Complejo deportivo .....	12
3.1.1.3 Espacios complementarios .....	12
3.1.1.4 Servicios auxiliares.....	12
3.1.1.5 Cancha .....	12
3.1.2 Clasificacion .....	12
3.1.2.1 Espacios deportivos convencionales.....	12
3.1.2.2 Espacios deportivos singulares .....	13
3.1.2.3 Áreas de actividad deportiva .....	13
3.2 Administracion de Horarios y Reservas .....	13
3.2.1 Administracion de Horarios .....	13
3.2.2 Reservas .....	14
3.2.1.1 Reservas regulares.....	15
3.2.1.2 Reservas provenientes de eventos .....	17
3.2.1.2 Reservas procedentes de dias de mantenimiento .....	17
3.2.3 Asignación de Recursos .....	17
3.2.3.1 Teoria de Colas .....	17
3.2.3.2 Elementos existentes en la teoría de colas.....	18
3.3 Cloud Computing.....	19
3.3.1 Definición .....	19
3.3.2 Características .....	20
3.3.3 Servicios Ofrecidos .....	20
3.3.3.1 Software como servicio (Saas) .....	20

3.3.3.2 Plataforma como servicio (Paas) .....	20
3.3.3.3 Infraestructura como servicio.....	20
Bibliografía .....	22

# CAPITULO I - INTRODUCCIÓN

## 1.1 Antecedentes

El problema de asignación y reservas de canchas en los complejos deportivos, asociaciones y clubes es bastante complejo ya que se busca determinar que canchas de cada disciplina deben asignarse a que clientes en qué fecha y varios aspectos que determinan los diferentes complejos deportivos, además la gran cantidad de información con que cuentan para realizar reservas.

Actualmente la ciudad de Cochabamba cuenta con un número importante de áreas deportivas entre complejos, asociaciones y pequeñas empresas que se dedican al flete de canchas. Estas áreas deportivas ofrecen diferentes disciplinas, precios de flete y calidad.

Los sistemas de administración de horarios tienen como misión fundamental ayudar a planificar horarios de forma precisa, ahorrar tiempo en la elaboración de los mismos y reducir las posibilidades de error que podrían presentarse cuando las personas intentan hacerlo por sí mismos.

Los sistemas de administración de horarios son considerados como una buena herramienta para la toma de decisiones oportunas, confiables y efectivas en cuanto a técnicas de planificación con el fin de garantizar el éxito, limitar el riesgo, reducir costos y aumentar las ganancias. De la misma forma estos sistemas proporcionan una variedad considerable de reportes para la empresa, almacenando grandes cantidades de información.

Los sistemas web o también conocidos como “Aplicaciones web” son aquellos que se alojan en servidores conectados a internet o una intranet (red local) que permiten acceder a ellos desde cualquier tipo de navegador. Su aspecto es muy similar a las páginas web que vemos normalmente pero en realidad los sistemas web tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Por otra parte, los sistemas web se han convertido en un factor importante para la vida de una empresa sea grande mediana o pequeña debido a que proporciona un medio libre de publicidad las 24 horas del día, que da como resultado el crecimiento de clientes y la buena imagen de la empresa. Asimismo llegan a reducir costos operativos.

El Cloud computing consiste en la posibilidad de ofrecer servicios a través de internet, de esta manera se puede tener todos los archivos e información en internet sin depender de la capacidad suficiente de almacenar información. Es una nueva forma de negocio más conocidos como e-buisness o negocios por internet .

Una aplicación de Cloud computing permite a los usuarios contar con la información en servidores con conexión a internet, de tal forma que el usuario solo necesita un navegador para poder acceder a dicha información, sin importar donde se encuentre.

## 1.2 Análisis del Problema

### 1.2.1 Definición del Problema

Deficiente manejo de información de reservas de canchas deportivas debido a la carencia de una administración precisa, óptima y confiable de horarios.

## **1.3 Objetivo del Proyecto**

### **1.3.1 Objetivo General**

Mejorar el manejo de información de la administración de horarios y reservas de las canchas deportivas a través del desarrollo de un sistema web para la administración de horarios y reservas para complejos, usando cloud computing.

### **1.3.2 Objetivos Específicos**

- Estudiar los procesos que comprenden en la administración de horarios y reservas de los complejos deportivos.
- Identificar las necesidades y requerimientos de los usuarios.
- Llevar a cabo la Implementación de la administración de horarios de complejos deportivos.
- Llevar a cabo la implementación de la administración de reservas.
- Permitir a los deportistas poder encontrar canchas a través del sistema web de planificación.
- Poner en productivo el sistema web en un servidor de la nube.

### **1.4 Áreas Involucradas**

- Sistemas de información
- Programación web

### **1.5 Justificación**

La revolución tecnológica que actualmente estamos viviendo bien podría ser la más profunda de nuestra historia. Los servicios convergen y pasan del mundo físico al mundo digital, siendo accesibles desde cualquier dispositivo. Un hecho relevante es que nuestros datos ya no residen en nuestros ordenadores sino en una Internet Global que adquiere entidad propia y se convierte en mucho más que una simple infraestructura de conexión: es la plataforma que ofrece servicio a millones de dispositivos inteligentes conectados a la red.[1]

Es lo que se conoce como Cloud Computing o informática en la nube de Internet, que permite que los consumidores, empresas o particulares, no se tengan que preocupar de cómo se provee el servicio que necesitan. Las empresas no podrán evitar este cambio si no quieren perder el tren del avance tecnológico, y esto implicará tomar decisiones sobre la dirección a seguir para mejorar sus negocios.[1]

El sistema web a desarrollar permitirá que los complejos cuenten con la información de horarios de manera más eficiente y rápida, de tal forma que optimizara el tiempo del proceso de reservas. Además, existiendo el riesgo de cometer errores en la planificación de reservas, el sistema las evitara de manera automática, eliminando la posibilidad de fallas.

Ya que en la actualidad contar con una página en internet representa una ventaja para las empresas, sin importar el tamaño de las mismas, un sistema de planificación de horarios y reservas vía web permitirá a los complejos deportivos contar con publicidad las 24 horas del día, beneficiando así a los complejos que no cuentan con estrategias de marketing.

En los países más desarrollados, está demostrado que la intensidad en la aplicación de nuevas tecnologías y la incorporación de un Sitio Web, está correlacionado de manera positiva con el incremento de las ventas, la productividad y el valor de mercado de las empresas. No importa el tamaño de la empresa, es de suma importancia el poder alcanzar a sus clientes de una manera masiva y sencilla para ellos.

Si bien es cierto que existen muchas redes sociales en las que te puedes anunciar de forma gratuita o pagada, el tener un Sitio Web eleva el prestigio propio de la empresa y permite a tus clientes incrementar el nivel de confianza hacia tu producto o servicio. Las redes sociales muestran resultados aleatorios que no siempre van de la mano con el comportamiento del consumidor final.

Dado que el sistema se encontrara en servidores web y disponible para su acceso, las personas podrán ver la disponibilidad de las canchas sin necesidad de ir en persona al complejo deportivo, representando así un beneficio para los deportistas ya que contarán con la información de canchas y horarios de varios complejos a través de internet.

### **1.6 Alcance**

- La aplicación será desarrollada para que funcione vía web.
- Para el desarrollo del proyecto se hará uso de la metodología scrum, con las adaptaciones necesarias dado que el proyecto será desarrollado por una persona.
- El sistema funcionara en los navegadores de Google Chrome v44, Firefox v39 y versiones superiores.
- El sistema podrá visualizarse en los sistemas operativos de Windows 7 y versiones superiores.
- La cancelación de reservas no se las realizara vía internet debido a que se debe pagar una licencia y a los bajos costos que algunos complejos deportivos piden por reservar una cancha.

## CAPITULO II - MARCO TEORICO

Este capítulo está dedicado a realizar una descripción completa de los elementos conceptuales que se usan en el proyecto. Define las características principales y justificación de cada elemento.

### 2.1 Sistema Administrativo

#### 2.1.1 Definicion

Definimos a los sistemas administrativos como la red de procedimientos relacionados de acuerdo a un esquema integrado tendientes al logro de los fines de una organización. Esto quiere decir que un conjunto de procedimientos relacionados y dentro de determinadas condiciones, constituyen un sistema.

#### 2.1.2 Sistema web

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

#### 2.1.3 Cloud Computing

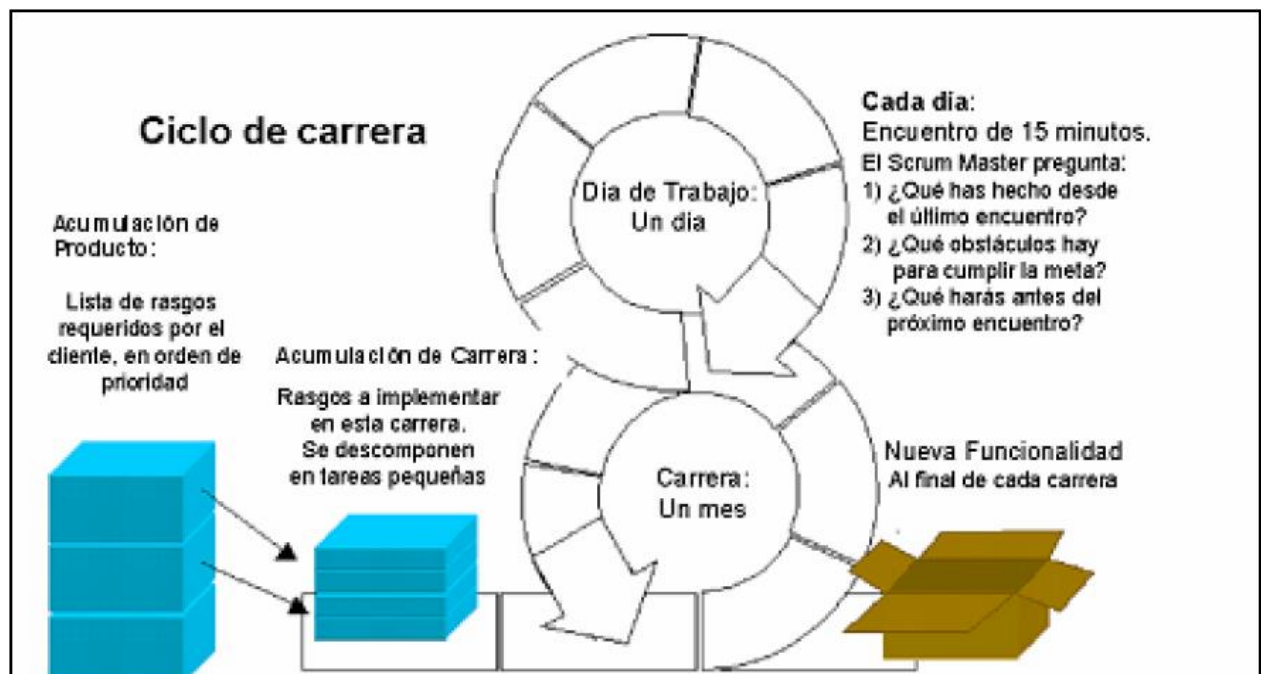
La computación en la nube, concepto conocido también bajo los términos servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, del inglés cloud computing, es un paradigma que permite ofrecer servicios de computación a través de Internet.

### 2.2 Metodología del Proyecto

#### 2.2.1 Metodologia Scrum

##### 2.2.1.1 Definicion

Scrum define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la impredecibilidad y el riesgo a niveles aceptables. El mismo surge en 1986, de un artículo de la Harvard Business Review titulado "The New New Product Development Game" de Hirotaka Takeuchi e Ikujiro Nonaka, que introducía las mejores prácticas más utilizadas en 10 compañías japonesas altamente innovadoras. A partir de ahí y tomando referencias al juego de rugby, Ken Schwaber y Jeff Sutherland formalizan el proceso conocido como Scrum en el año 1995.



*Figura 1: Metodologia scrum*

El marco técnico de scrum está formado por:

Roles:

- El equipo scrum.
- El dueño del producto.
- El Scrum Master.

Artefactos:

- Pila del producto.
- Pila del sprint.
- Incremento.
- Sprint.

Eventos

- Reunión de planificación del sprint.
- Scrum diario.
- Revisión del sprint.
- Retrospectiva del sprint.

En donde lo mas importante es el sprint, que dura aproximadamente entre 2 a 4 semanas.

### **2.2.1.2 Caracteristicas**

- Enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas
- Hace uso de Equipos auto-dirigidos y auto-organizados
- Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.[2]
- Desarrollo de software iterativos incrementales basados en prácticas ágiles
- Iteraciones de treinta días; aunque se pueden realizar con mas frecuencia, estas iteraciones, conocidas como Sprint
- Dentro de cada Sprint se denomina al Scrum Master al Líder de Proyecto quien llevará a cabo la gestión de la iteración
- Se convocan diariamente un “Scrum Daily Meeting” el cual representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener realimentación sobre las tareas de los recursos y los obstáculos que se presentan. En la cual se responden preguntas como: ¿Qué has hecho desde el ultimo encuentro? ¿Qué obstaculos hay para cumplir la meta? ¿Qué haras antes del proximo encuentro?

### **2.2.1.3 Justificacion**

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminado el trabajo innecesario.
- Atención continua a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

## **2.3 Plataforma de desarrollo**

### **2.3.1 Lenguaje de Programación Javascript**

#### **2.3.1.1 Definición**

JavaScript(a veces abreviado como JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Es un lenguaje script



multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. Leer más sobre JavaScript.

JavaScript no debe ser confundido con el lenguaje de programación Java. Java es una marca registrada de Oracle en Estados Unidos y otros países.

Contrariamente a la falsa idea popular, JavaScript no es "Java interpretativo". En pocas palabras, JavaScript es un lenguaje de programación dinámico que soporta construcción de objetos basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje.[6]

### **2.3.1.2 Características**

- Imperativo y estructurado.
- Tipado dinámico.
- Puede funcionar como lenguaje procedimental y como orientado a objetos.
- Evaluación del código en tiempo de ejecución.
- Funcional.
- Basado en prototipos.

### **2.3.1.3 Justificación**

- Es el lenguaje de programación de los navegadores web (todos los más importantes lo soportan y lo tienen activado por defecto: Firefox, Chrome, IE, Opera, Safari...), lo que lo convierten en el lenguaje más popular en Internet.
- Hay una auténtica competición entre los navegadores para optimizar sus motores y dar mejor soporte a JavaScript y que su código se ejecute más rápidamente. JavaScript por tanto es cada vez más estable y tiene mejor rendimiento.
- Es muy potente y expresivo, con sintaxis que guarda similitudes con otros lenguajes muy populares, pero con características particulares.
- Puede interaccionar con otras tecnologías populares como Flash, Java, PHP, etc.
- Existen miles de bibliotecas para trabajar con javascript

### **2.3.2 Framework Meteor**

#### **2.3.2.1 Definición**

Meteor es un framework para desarrollo de aplicaciones web, el cual utiliza Node.js en su núcleo, permitiéndonos de esta manera programar en un mismo lenguaje (JavaScript) tanto del lado del cliente como del servidor. Lo que es más, Meteor también es capaz de compartir código entre ambos entornos. Entre sus principales características esta el poder crear aplicaciones "Realtime" de una manera realmente rápida y sencilla.

Meteor se encuentra entre la base de datos de la aplicación y la interfaz de usuario y se asegura de que ambos se mantengan en sincronía.[5]

La arquitectura que usa Meteor es la siguiente:

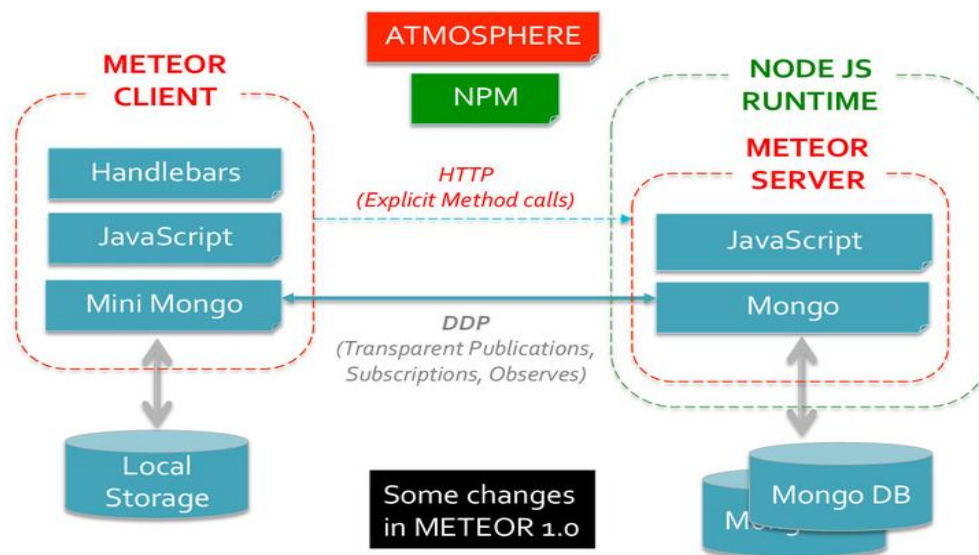


Figura 2: Arquitectura de Meteor

### 2.3.2.2 Características

- **Transferencia de datos.** Meteor no envía HTML sobre la red. El servidor envía los datos y permite al cliente visualizarlo.
- **Un solo lenguaje.** Meteor permite escribir tanto el cliente como el servidor de tu aplicación en JavaScript.
- **Base de datos en todas partes.** Puedes usar los mismos métodos para acceder a la base de datos tanto desde el cliente, como desde el servidor.
- **Compensación de latencia.** En el cliente, Meteor presolicita los datos y simula modelos para hacer parecer que la llamada al servidor es instantánea.
- **Reactividad en Full Stack.** En Meteor, el tiempo real es por defecto. Todas las capas de la aplicación, desde la base de datos hasta la plantilla, se actualizan de forma automática cuando es necesario.
- **Adopta el ecosistema.** Meteor es código abierto, se integra con herramientas y frameworks de código abierto.
- **La simplicidad es igual a la productividad.** La mejor manera de hacer que algo parezca simple es hacerlo simple. La principal funcionalidad de Meteor es clara, crear bellas APIs.

### 2.3.2.3 Justificación

- **Buena base tecnológica.** La parte de servidor de Meteor funciona sobre Node.js pero eliminando problemas como el famoso “callback hell” utilizando Fibers y actualmente utiliza MongoDB como base de datos utilizando oplogtailoring para monitorizar los cambios de forma eficiente.
- **Facil de aprender .** Javascript en todas partes. Solo necesitas aprender un lenguaje.
- **Inspeccion de codigo.** En el lado del cliente, es posible utilizar cualquiera de las herramientas que tenemos a nuestro alcance ofrecidas por los distintos navegadores. Meteor tiene soporte para debugging\_ en el lado del servidor y WebStorm ha lanzado hace poco una versión totalmente compatible con Meteor, debugging incluido.
- **Simplicidad en el cliente.** A diferencia de otros frameworks como Angular, Meteor no mezcla la lógica de la aplicación con el HTML. El sistema utilizado por Meteor, Blaze no incluye en el HTML javascript, lógica de bindings ni atributos extraños como data- o ng-. No es necesario inventar tus propias etiquetas HTML.

- **Seguridad.** Meteor tiene medidas de seguridad de serie frente a muchos de los problemas de seguridad más frecuentes.
- **Pruebas.** El framework oficial de Testing de Meteor es Velocity y nos permite integrar las funcionalidades propias del tiempo real con Jasmine o Mocha. Los tests se ejecutan automáticamente cuando guardamos los cambios y el estado de nuestros tests se indica en directo con un punto verde o rojo en la parte superior derecha de la aplicación.
- **Soporte Movil.** Las aplicaciones móviles se pueden generar usando el mismo código que la versión web usando PhoneGap/Cordova.

### 2.3.3 Framework Semantic UI

#### 2.3.3.1 Definición

Semantic UI es un framework de aplicaciones front end que es compuesto por un conjunto de especificaciones para el intercambio de elementos de la interfaz entre los desarrolladores.

El vocabulario utilizado (para las clases e ID) es mucho más claro en comparación con las alternativas que reduce la curva de aprendizaje.

Semantic UI cuenta con muchos estilos para los elementos HTML, los elementos de la interfaz de usuario y escenarios (botones, formularios, tablas, acordeón, navegación, comentando, etc.).[4]

#### 2.3.3.2 Características

- **Html conciso.** Las clases utilizan la sintaxis de lenguaje natural como sustantivo / relaciones modificadoras, orden de las palabras, y la pluralidad de vincular conceptos intuitivamente.
- **Javascript Intuitivo.** Semantic UI utiliza frases simples llamados comportamientos que desencadenan funcionalidad.
- **Diseño Responsive.** Semantic UI está diseñado completamente con medidas “em” haciendo sensible dimensionar componentes.
- **Socios.** Semantic UI tiene integraciones con React, Angular, Meteor, Ember y muchos otros Frameworks para ayudar a organizar la interfaz de usuario.

#### 2.3.3.3 Justificación

- Semantic cuenta con más de 3000 variables CSS.
- Compuesta por 50 definiciones de elementos UI prediseñados que puedes utilizar para aplicar en el diseño de una página o aplicación.
- Es un proyecto Open Source en el cual están involucrados múltiples diseñadores y desarrolladores quienes han puesto un granito de arena para hacer posible este proyecto que actualmente cuenta con más de 3800 Commits.
- Semantic UI al ser un proyecto Open Source es completamente gratuito.
- Semantic UI esta diseñado para ser completamente responsive.
- Cuenta con una buena documentacion.

## 2.4 Administrador de Base de Datos

### 2.4.1 MongoDB

#### 2.4.1.1 Definición

MongoDB es un sistema de base de datos **multiplataforma** orientado a **documentos**, de esquema libre. Como ya os expliqué, esto significa que cada entrada o registro puede tener un esquema de datos diferente, con atributos o “columnas” que no tienen por qué repetirse de un registro a otro. Está escrito en C++, lo que le confiere cierta cercanía al *bare metal*, o recursos de hardware de la máquina, de modo que es bastante rápido a la hora de ejecutar sus tareas. Además, está licenciado como GNU AGPL 3.0, de modo que se trata de un software de licencia libre. Funciona en sistemas operativos Windows, Linux, OS X y Solaris.

Los distintos documentos se almacenan en formato **BSON**, o Binary JSON, que es una versión modificada de JSON que permite búsquedas rápidas de datos. BSON guarda de forma explícita las longitudes de los campos, los índices de los vectores, y demás información útil para el escaneo de datos. Es por esto que, en algunos casos, el mismo documento en BSON ocupa un poco más de espacio de lo que ocuparía de estar almacenado directamente en formato JSON. Pero una de las ideas claves en los sistemas NoSQL es que el almacenamiento es barato, y es mejor aprovecharlo si así se introduce un considerable incremento en la velocidad de localización de información dentro de un documento.[3]

#### 2.4.1.2 Terminología básica de MongoDB

En MongoDB, cada registro o conjunto de datos se denomina **documento**. Los documentos se pueden agrupar en **colecciones**, las cuales se podría decir que son el equivalente a las tablas en una base de datos relacional (sólo que las colecciones pueden almacenar documentos con muy diferentes formatos, en lugar de estar sometidos a un esquema fijo). Se pueden crear **índices** para algunos atributos de los documentos, de modo que MongoDB mantendrá una estructura interna eficiente para el acceso a la información por los contenidos de estos atributos.

#### 2.4.1.3 Arquitectura de MongoDB

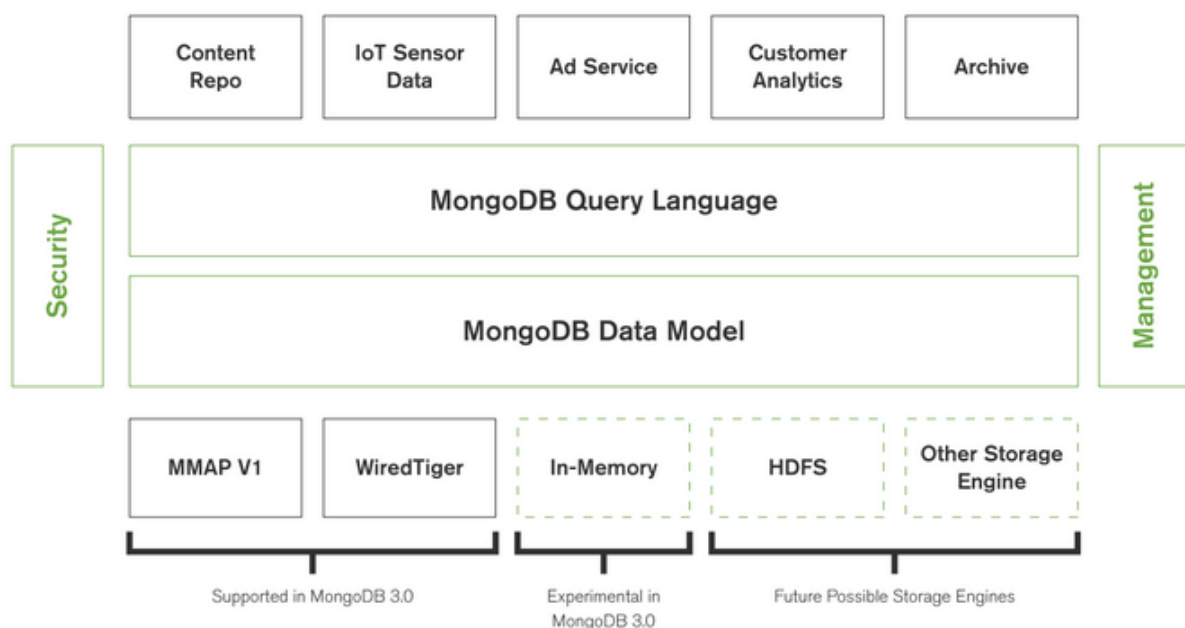


Figura 3: Arquitectura de Mongo DB

#### 2.4.1.4 Características

- **Modelo de datos flexible.** Bases de datos NoSQL surgió para hacer frente a los requisitos de los datos vemos que domina las aplicaciones modernas. Un modelo de datos flexible hace que sea fácil de almacenar y combinar los datos de cualquier estructura, y permita la dinamización del esquema sin tiempo de inactividad.
- **La escalabilidad elástica.** Bases de datos NoSQL fueron todas construidas con un enfoque en la escalabilidad, por lo que todos ellos incluyen algún tipo de sharding o partición, permitiendo que la base de datos a escala de salida en el hardware de los productos básicos, lo que permite un crecimiento casi ilimitado.
- **Alto Rendimiento.** Bases de datos NoSQL están diseñados para ofrecer un gran rendimiento, medido en términos de rendimiento y latencia en cualquier escala.

#### **2.4.1.5 Justificación**

- Da respuesta a la necesidad de almacenamiento de todo tipo de datos: estructurados, semi estructurados y no estructurados.
- Tiene un gran rendimiento en cuanto a escalabilidad y procesado de la información.
- Puede procesar la gran cantidad de información que se genera hoy en día (millones de usuarios utilizando las mismas aplicaciones, redes sociales que crecen sin parar, internet de las cosas, cloud computing, etc).
- Se adapta a las necesidades actuales de las aplicaciones (millones de usuarios, miles de peticiones por segundo, etc).
- Permite a las empresas ser más ágiles y crecer más rápidamente, crear nuevos tipos de aplicaciones/productos, mejorar la experiencia del cliente y reducir el tiempo de manufacturado o desarrollo del producto, reduciendo así los costes.
- Está orientada a documentos. Lo que quiere decir que en un único documento es capaz de almacenar toda la información necesaria que define un producto, un cliente, etc, aceptando todo tipo de datos (incluidos arrays y otros subdocumentos). Todo ello sin tener que seguir un esquema predefinido.
- Permite adaptar el esquema de la base de datos a las necesidades de la aplicación rápidamente, disminuyendo el tiempo y coste de la puesta en producción de la misma. Esto es así porque permite modificar el esquema desde el código de la aplicación, sin tener que realizar labores de administración de la base de datos como estaríamos obligados a hacer con las bases de datos relacionales.
- Da a los desarrolladores todas las funcionalidades que tienen las bases de datos relacionales (como índices sobre campos secundarios, un completo lenguaje para realizar las consultas, etc).

## CAPITULO III – AREAS DE APLICACION

En este capítulo se realiza una definición sobre complejos deportivos, sus características y tipos. Se hace una diferencia entre instalación deportiva y complejo deportivo. Se definen conceptos secundarios sobre complejos deportivos, como instalación deportiva, cancha, servicios auxiliares, etc. Se hace una descripción completa de los procesos de registro de campos deportivos, realización de una reserva, etc, así como el diagrama de flujo de proceso para cada una de ellas.

### 3.1 Instalación deportiva

Una instalación deportiva es un recinto o una construcción provista de los medios necesarios para el aprendizaje, la práctica y la competición de uno o más deportes. Incluyen las áreas donde se realizan las actividades deportivas, los diferentes espacios complementarios y los de servicios auxiliares. Las instalaciones deportivas se componen de uno o más espacios deportivos específicos para un tipo de deporte.

Una instalación deportiva puede tener un solo espacio deportivo o varios, cada uno destinado a un deporte diferente. Así, por ejemplo, un estadio de fútbol puede contener una pista de atletismo, y albergar en el sótano un gimnasio y una sala polideportiva; en este caso se habla de una instalación deportiva con cuatro espacios deportivos diferentes.

Ejemplos de instalaciones deportivas son los estadios, los pabellones deportivos, velódromos, pistas de tenis, piscinas, canales de remo y piragüismo, marinas deportivas, estaciones de esquí, circuitos de bicicletas, campos de tiro, de hípica, de golf, etc.



Figura 4 : Ciclo de vida de una instalación deportiva

Fuente:

#### 3.1.1 Terminología

##### 3.1.1.1 Espacio deportivo

Es el espacio donde se desarrolla una actividad deportiva específica, por ejemplo una sala de artes marciales, la pista de carreras en un estadio de atletismo, un campo de tiro al plato o una pista de snowboard.

### 3.1.1.2 Complejo deportivo

Se trata de dos o más instalaciones deportivas ubicadas en un recinto común y con fácil acceso entre cada una de sus partes; funcionan independientemente entre sí y se conocen generalmente bajo una misma denominación. El llamado Anillo Olímpico de Montjuïc, en Barcelona, es un complejo deportivo que incluye varias instalaciones: el Estadio Olímpico Lluís Companys (atletismo y fútbol), el Palau Sant Jordi (polideportivo), las Piscinas Bernat Picornell (natación), un campo de béisbol y otro de hockey hierba.

### 3.1.1.3 Espacios complementarios

Sirven para dar apoyo a las actividades deportivas desarrolladas en los diferentes espacios deportivos; el deporte no es realizado en estos espacios. Ejemplos: vestuarios, aseos, primeros auxilios, control antidopaje, almacenes de material deportivo, gradas, etc.

### 3.1.1.4 Servicios auxiliares

No están relacionados con las actividades deportivas. Pueden ser de diversa índole: cafeterías, bares, guarderías, tiendas, servicio médico, cuartos de máquinas, de calderas, etc.

### 3.1.1.5 Cancha

El concepto de “cancha” tiene origen quechua. Es posible, según los expertos en cuestiones lingüísticas, distinguir entre dos raíces etimológicas de la palabra: kamcha y kancha. En el primer caso, se refiere a las habas ya tostadas y al maíz que los habitantes de la región sudamericana acostumbran ingerir.

En cambio, la palabra cancha inspirada en kancha (traducida como “recinto”) posee una mayor amplitud de acepciones. El vocablo se utiliza para referirse al espacio reservado para la práctica de distintos deportes o para la organización de espectáculos. Entre los ejemplos de cancha tenemos : fútbol-sala, fútbol, basket, etc.

## 3.1.2 Clasificación

Una instalación deportiva se clasifica de acuerdo a sus espacios deportivos. Estos pueden ser de tres tipos:

### 3.1.2.1 Espacios deportivos convencionales

Dan servicio a las prácticas deportivas más comunes, y atienden a referentes reglamentados con dimensiones normalizadas, aunque no siempre se ajustan a ellas.

Hay seis tipos de espacios convencionales:

**Campos:** generalmente son de forma rectangular, al aire libre y con delimitación y marcación clara. Superan los 1500 m<sup>2</sup> de superficie. Ejemplos: campos polideportivos, de fútbol, fútbol 7, rugby, hockey sobre hierba, béisbol, zonas de lanzamiento de atletismo.

**Espacios longitudinales:** espacios en los que la actividad se realiza siguiendo un recorrido fijo y delimitado. Ejemplos: pistas de atletismo (de 200 m, 300 m o 400 m), rectas de saltos atléticos, velódromos, patinódromos.

**Pistas:** son como los campos, de forma rectangular y con delimitación y marcación clara, generalmente están al aire libre, aunque las hay cubiertas (en pabellones). Su superficie es menor a 1500 m<sup>2</sup>. Ejemplos: pistas polideportivas, de baloncesto, voleibol, vóley playa, tenis, bádminton, patinaje sobre ruedas, patinaje sobre hielo, hockey sobre hielo, etc.

**Pistas con pared:** son como las pistas pero con una o más paredes que son necesarias para el desarrollo de la actividad deportiva. Ejemplos: pistas de frontón, squash, padel, etc.

**Salas:** son espacios cubiertos de poca superficie donde se pueden realizar diferentes actividades deportivas y que no tienen los requerimientos dimensionales y de marcación de las pistas. Ejemplos: las salas de artes marciales, esgrima, gimnasia, halterofilia, etc.



**Piscinas:** cuerpos cubiertos de agua para la práctica de deportes acuáticos, pueden estar al aire libre o cubiertas. Piscinas de natación (de 25 m, 33 m o 50 m), saltos, waterpolo.

### 3.1.2.2 Espacios deportivos singulares

Son espacios más específicos que suelen estar sujetos a unos requerimientos espaciales.

Representativos de este tipo son:

**Campos de golf:** de minigolf, de 18 hoyos, de 9 hoyos.

**Campos de tiro:** para tiro con arco, tiro de precisión o tiro al plato.

**Canal de regatas:** para remo, piragüismo en aguas tranquilas o piragüismo en eslalon.

**Circuito de bicicleta:** circuitos cerrados y carril bici.

**Circuitos de carrera a pie.**

**Circuitos de motor:** circuitos de velocidad (para autos o [[Motociclismo de velocidad|motocicletas), de motocross, de kart

**Espacios de hípica:** pistas de doma, de saltos, hipódromos, campos de polo, etc.

**Estaciones de esquí:** para esquí de fondo, esquí alpino, snowboard, etc.

### Rocódromos

**Zonas de juegos populares o tradicionales:** boleras, billares, pistas de petanca, etc.

### 3.1.2.3 Áreas de actividad deportiva

Se distinguen por la indefinición de sus límites y por el medio natural en el que la práctica físicodeportiva tiene lugar: acuático, aéreo o terrestre.

Según el medio natural en el que la actividad deportiva es realizada, se distinguen tres tipos:

**Áreas de actividad acuática:** zonas de barranco (ráfting, piragüismo en aguas bravas), de actividades subacuáticas, surf, vela, esquí acuático, de remo y piragüismo (en embalses, lagos o bahías), etc.

**Áreas de actividad aéreas:** zonas de paracaidismo, aeromodelismo, vuelo sin motor, vuelo aerostático, parapente, ala delta, etc.

**Áreas de actividad terrestres:** zonas de escalada, espeología, senderos, campo a través (atletismo, BTT, ecuestre, esquí), rutas ecuestres, campos de caza, etc.

## 3.2 Administracion de Horarios y Reservas

### 3.2.1 Administracion de Horarios

Administracion de horarios es el proceso de desarrollo, mantenimiento y comunicación de los horarios de tiempo y de recursos.

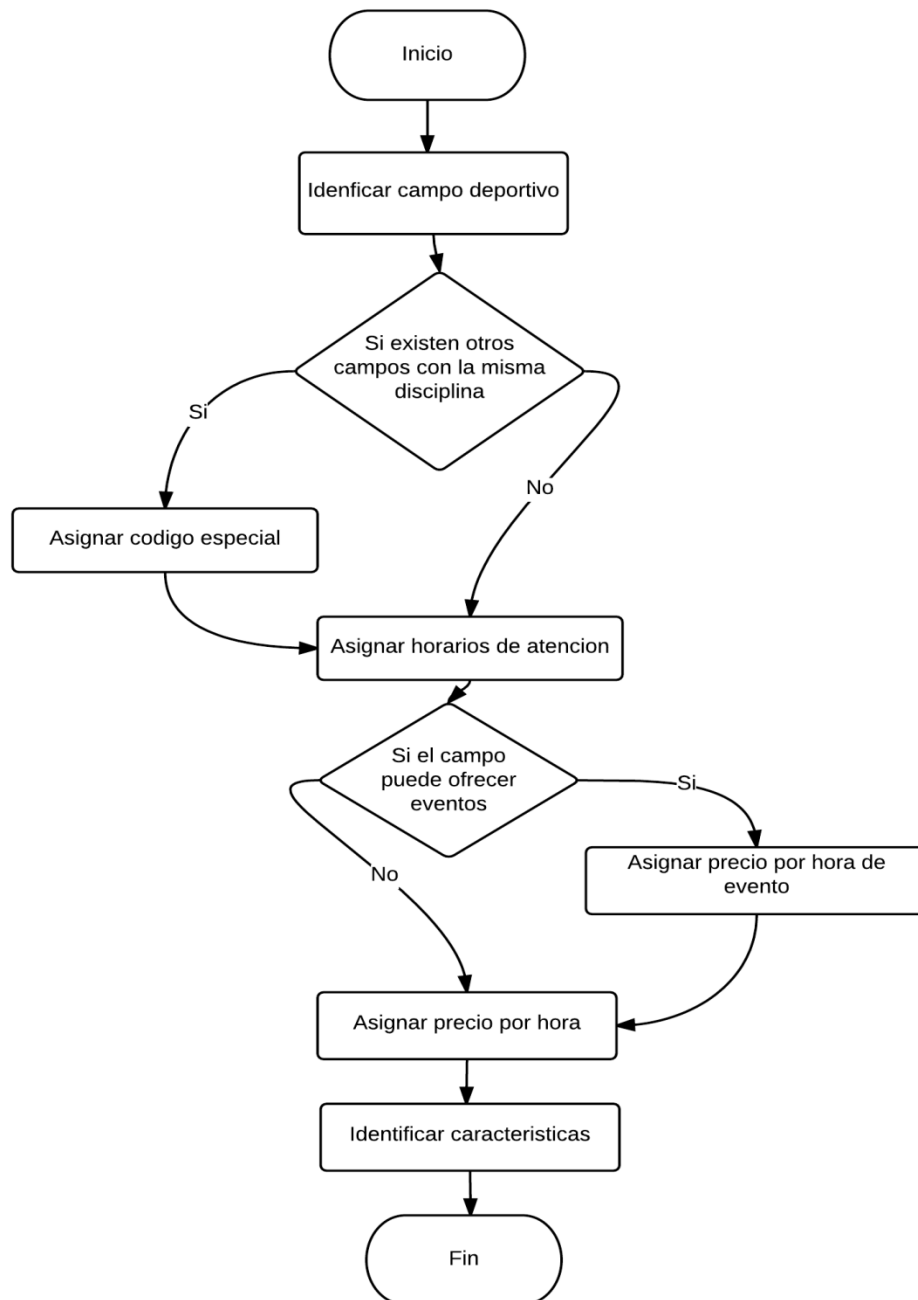
Para hacer esto posible, los complejos deportivos realizan un registro de cada campo existente en sus instalaciones.

El proceso que se lleva a cabo para llevar un control de horarios es: El administrador del complejo identifica a cada campo con un código, ya que pueden existir varios campos con la misma disciplina deportiva, asimismo identifica las características del campo, como pueden ser: tipo de suelo, el precio por cada hora de reserva. Una de las características mas importantes es el de la hora de atención del campo deportivo, el cual será definido por el administrador del complejo al momento de hacer el registro del campo.

La división de horarios que se realiza por día, es decir, el número de veces máximo por día que puede ser reservado un campo deportivo es segmentado por horas enteras, ejemplo: 8:00 a 9:00, 9:00 a 10:00, etc.

El diagrama de flujo del proceso se puede observar en la imagen inferior.





*Figura 6 : Diagrama de flujo para el registro de una campo deportivo*

*Fuente : Elaboracion propia*

### 3.2.2 Reservas

En principio, las reservas son beneficios obtenidos por la empresa y que no han sido distribuidos entre sus propietarios. Pero este concepto solamente es válido desde una perspectiva amplia, ya que se puede hacer una subdivisión de las distintas clases de reservas en función de su origen.

Así, se diferencian tres grandes bloques de reservas:

- Reservas regulares.
- Reservas provenientes de eventos.
- Reservas procedentes de días de mantenimiento.

#### **3.2.1.1 Reservas regulares**

La característica común de estas reservas es que se originan cuando la empresa ha tenido beneficios. Si esto ha sucedido, la cuenta que expresa el beneficio tendrá un saldo acreedor y cuando se acuerde su traspaso a cualquier tipo de las reservas anteriormente citadas.

El proceso de este tipo de reserva, comienza cuando un cliente o usuario del complejo deportivo realiza una petición de uso de algún campo deportivo, especificando el campo deportivo y horario que desea utilizar. Un elemento importante a tomar en cuenta es que los complejos deportivos manejan una lista de clientes frecuentes por campo deportivo, el cual prioriza a los clientes que más veces han usado el campo deportivo, formando una cola de clientes prioritarios. Así, si no existe una prereserva por algún cliente prioritario, la reserva del campo deportivo se la realiza de manera normal, pero si existiese, el campo deportivo queda reservado para dicho cliente prioritario. Una vez realizada la reserva, de acuerdo al número de horas de la reserva, se calcula el precio total de la reserva, de acuerdo a la siguiente fórmula  $Pt = N_{horas} * Phora$ , esto sirve para el reporte diario de reservas y para la emisión de facturas.

El proceso de reservas regulares se observa en la *Figura 7*.

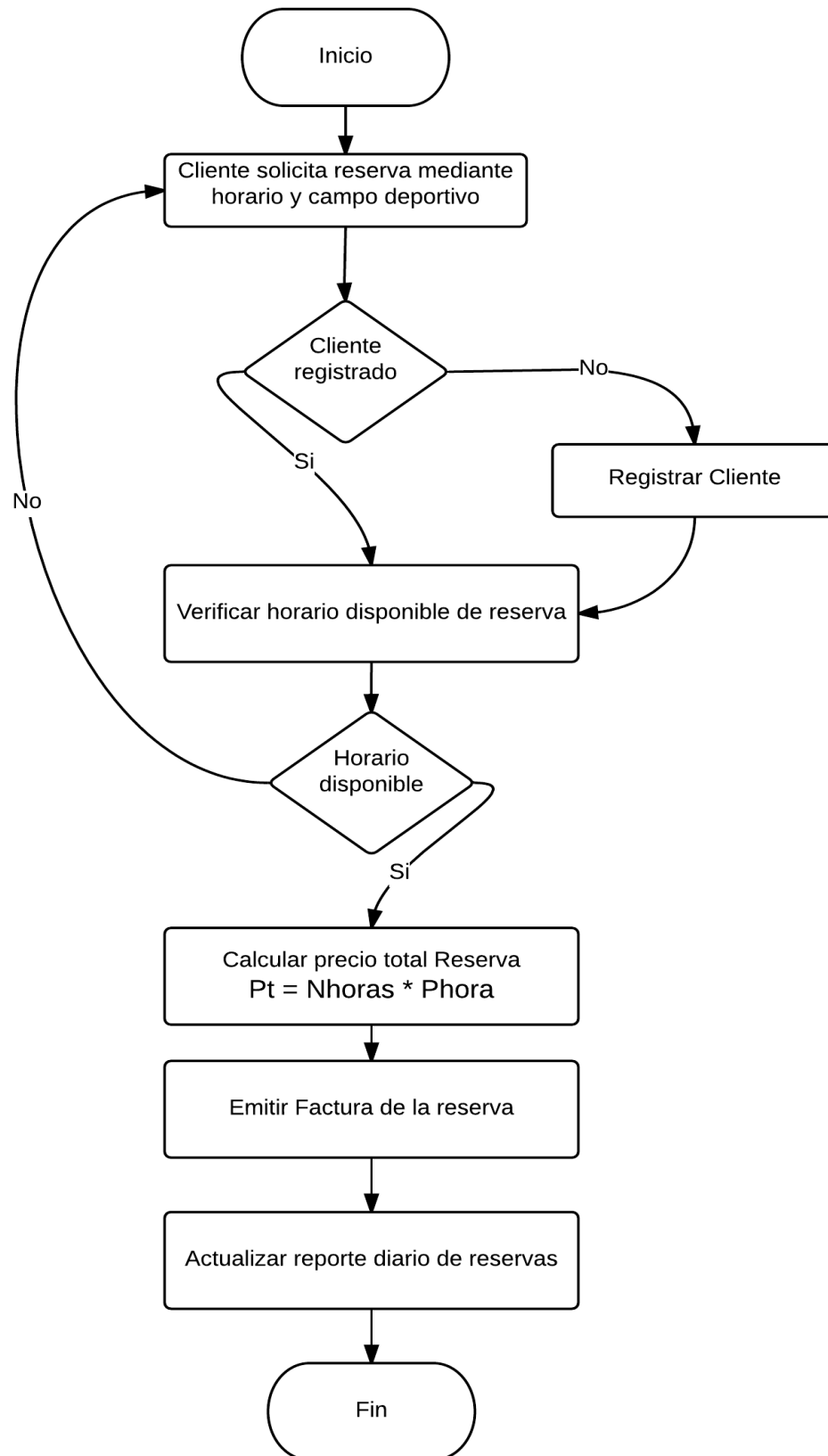


Figura 8 : Diagrama de flujo para una reserva

Fuente : Elaboracion propia

### **3.2.1.2 Reservas provenientes de eventos**

Las reservas de tipo evento son parecidas a las reservas regulares, con la diferencia de que por este tipo de reservas, se cobra un precio superior.

Para poder reservar un evento en un campo deportivo, se sigue los siguientes pasos:

El Cliente solicita un evento en un campo deportivo. Posteriormente el encargado de reservas del complejo deportivo notifica al administrador que dicho campo es reservado para un evento (el cual tendrá un precio superior al de una reserva regular). El administrador prepara un documento, el cual representa un contrato para la reserva del evento, que entre los datos más importantes de dicho contrato están: el nombre del administrador, nombre del cliente con sus respectivas firmas, el campo deportivo a reservar, horarios de reserva y el precio por la reserva. Una copia del contrato es entregado al cliente y la copia original es guardada en los documentos del complejo deportivo.

Una vez las partes firman el contrato, se procede a registrar la reserva en los horarios establecidos. (Falta diagrama de flujo)

### **3.2.1.2 Reservas procedentes de días de mantenimiento**

Este tipo de reservas se dan cuando un campo deportivo necesita de un mantenimiento, ya sea de limpieza o refacción del campo deportivo.

La característica principal de este tipo de reserva es que no genera ningún ingreso económico para el complejo, más al contrario, es una pérdida económica pero necesaria para mantener en buen estado el campo deportivo.

Esta reserva la autoriza el administrador del complejo deportivo.

### **3.2.3 Asignación de Recursos**

En las ciencias económicas y empresariales, la expresión Asignación de Recursos designa la forma como una determinada economía distribuye sus recursos (llamados factores de producción) por los diversos usos posibles con el fin de producir un determinado conjunto de productos o servicios finales.

El propósito de cualquier economía es hacer que una asignación de recursos lo más eficiente para que sea posible maximizar el rendimiento final con el consumo mínimo de recursos, es decir, para maximizar la eficiencia y la productividad.

#### **3.2.3.1 Teoría de Colas**

La teoría de colas es el estudio matemático de las colas o líneas de espera dentro de un sistema. Esta teoría estudia factores como el tiempo de espera medio en las colas o la capacidad de trabajo del sistema sin que llegue a colapsarse. Dentro de las matemáticas, la teoría de colas se engloba en la investigación de operaciones y es un complemento muy importante a la teoría de sistemas y la teoría de control. Se trata así de una teoría que encuentra aplicación en una amplia variedad de situaciones como negocios, comercio, industria, ingenierías, transporte y logística o telecomunicaciones.

En el caso concreto de la ingeniería, la teoría de colas permite modelar sistemas en los que varios agentes que demandan cierto servicio o prestación, confluyen en un mismo servidor y, por lo tanto, pueden registrarse esperas desde que un agente llega al sistema y el servidor atiende sus demandas. En este sentido, la teoría es muy útil para modelar procesos tales como la llegada de datos a una cola en ciencias de la computación, la congestión de red de computadoras o de telecomunicación, o la implementación de una cadena productiva en la ingeniería industrial.

En el contexto de la informática y de las tecnologías de la información y la comunicación las situaciones de espera dentro de una red son más frecuentes. Así, por ejemplo, los procesos enviados a un servidor para su ejecución forman colas de espera mientras no son atendidos; la información solicitada, a través de Internet, a un servidor Web puede recibirse con demora debido a la congestión en la red; también se

puede recibir la señal de línea de la que depende nuestro teléfono móvil ocupada si la central está colapsada en ese momento, etc.

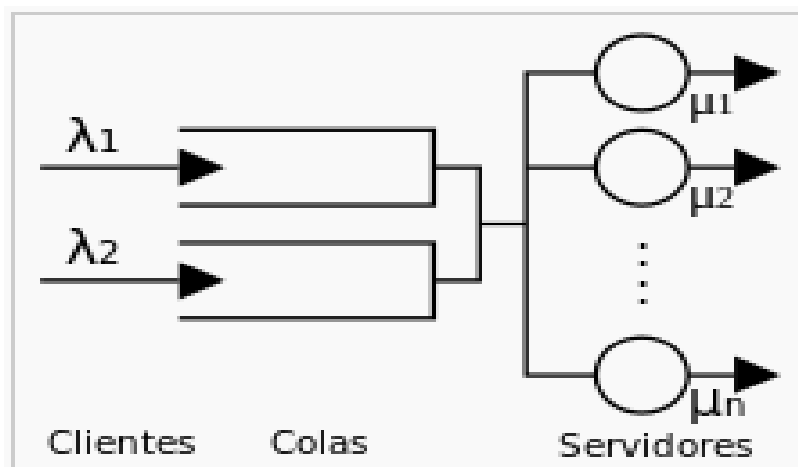


Figura 9 : Teoría de colas

Fuente : Wikipedia

### 3.2.3.2 Elementos existentes en la teoría de colas

- Proceso básico de colas: Los clientes que requieren un servicio se generan en una fase de entrada. Estos clientes entran al sistema y se unen a una cola. En determinado momento se selecciona un miembro de la cola, para proporcionarle el servicio, mediante alguna regla conocida como disciplina de servicio. Luego, se lleva a cabo el servicio requerido por el cliente en un mecanismo de servicio, después de lo cual el cliente sale del sistema de colas.

- Fuente de entrada o población potencial: Una característica de la fuente de entrada es su tamaño. El tamaño es el número total de clientes que pueden requerir servicio en determinado momento. Puede suponerse que el tamaño es infinito o finito.

- Cliente: Es todo individuo de la población potencial que solicita servicio como por ejemplo una lista de trabajo esperando para imprimirse.

- Capacidad de la cola: Es el máximo número de clientes que pueden estar haciendo cola (antes de comenzar a ser servidos). De nuevo, puede suponerse finita o infinita.

- Disciplina de la cola: La disciplina de la cola se refiere al orden en el que se seleccionan sus miembros para recibir el servicio. Por ejemplo, puede ser:

FIFO (first in first out) primero en entrar, primero en salir, según la cual se atiende primero al cliente que antes haya llegado.

LIFO (last in first out) también conocida como pila que consiste en atender primero al cliente que ha llegado el último.

RSS (random selection of service) que selecciona los clientes de manera aleatoria, de acuerdo a algún procedimiento de prioridad o a algún otro orden.

Processor Sharing – sirve a los clientes igualmente. La capacidad de la red se comparte entre los clientes y todos experimentan con eficacia el mismo retraso.

- Mecanismo de servicio: El mecanismo de servicio consiste en una o más instalaciones de servicio, cada una de ellas con uno o más canales paralelos de servicio, llamados servidores.

- Redes de colas. Sistema donde existen varias colas y los trabajos fluyen de una a otra. Por ejemplo: las redes de comunicaciones o los sistemas operativos multitarea.

- Cola: Una cola se caracteriza por el número máximo de clientes que puede admitir. Las colas pueden ser finitas o infinitas.

- El proceso de servicio: Define cómo son atendidos los clientes.

### 3.3 Cloud Computing

#### 3.3.1 Definición

La computación en la nube, conocida también como servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos (del inglés cloud computing), es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.[10]

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan.

La computación en la nube son servidores desde Internet encargados de atender las peticiones en cualquier momento. Se puede tener acceso a su información o servicio, mediante una conexión a internet desde cualquier dispositivo móvil o fijo ubicado en cualquier lugar.

El concepto de la computación en la nube empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon AWS, Microsoft y otros que construyeron su propia infraestructura. De entre todos ellos emergió una arquitectura: un sistema de recursos distribuidos horizontalmente, introducidos como servicios virtuales de TI escalados masivamente y manejados como recursos configurados y mancomunados de manera continua.

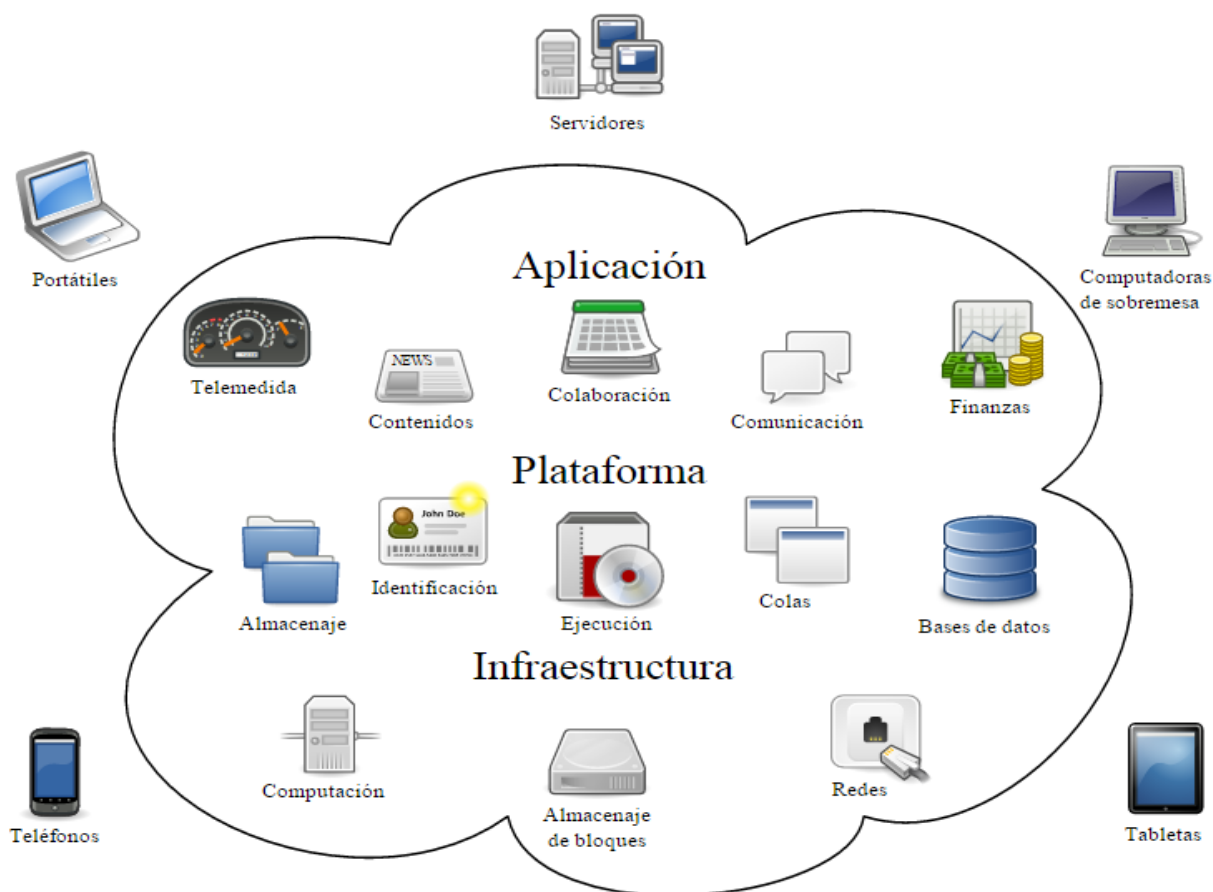


Figura x: Esquema de Cloud computing

Fuente : Wikipedia

### 3.3.2 Características

La computación en nube presenta las siguientes características clave:

**Costo:** los proveedores de computación en la nube afirman que los costes se reducen. Un modelo de prestación pública en la nube convierte los gastos de capital en gastos de funcionamiento.

**Rendimiento:** Los sistemas en la nube controlan y optimizan el uso de los recursos de manera automática, dicha característica permite un seguimiento, control y notificación del mismo. Esta capacidad aporta transparencia tanto para el consumidor o el proveedor de servicio.

**Seguridad:** puede mejorar debido a la centralización de los datos. La seguridad es a menudo tan bueno o mejor que otros sistemas tradicionales, en parte porque los proveedores son capaces de dedicar recursos a la solución de los problemas de seguridad que muchos clientes no pueden permitirse el lujo de abordar.

**Mantenimiento:** en el caso de las aplicaciones de computación en la nube, es más sencillo, ya que no necesitan ser instalados en el ordenador de cada usuario y se puede acceder desde diferentes lugares.

### 3.3.3 Servicios Ofrecidos

#### 3.3.3.1 Software como servicio (SaaS)

El software como servicio (en inglés software as a service, SaaS) se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, por-demanda, vía multitenencia —que significa una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes. Las aplicaciones que suministran este modelo de servicio son accesibles a través de un navegador web -o de cualquier aplicación diseñada para tal efecto- y el usuario no tiene control sobre ellas, aunque en algunos casos se le permite realizar algunas configuraciones. Esto le elimina la necesidad al cliente de instalar la aplicación en sus propios computadores, evitando asumir los costos de soporte y el mantenimiento de hardware y software.

#### 3.3.3.2 Plataforma como servicio (Paas)

La capa del medio, que es la plataforma como servicio (en inglés platform as a service, PaaS), es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno conteniendo una pila básica de sistemas, componentes o APIs preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby). Las ofertas de PaaS pueden dar servicio a todas las fases del ciclo de desarrollo y pruebas del software, o pueden estar especializadas en cualquier área en particular, tal como la administración del contenido.

#### 3.3.3.3 Infraestructura como servicio

La infraestructura como servicio (infrastructure as a service, IaaS) -también llamada en algunos casos hardware as a service, HaaS)<sup>7</sup> se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red. Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran (por ejemplo a través de la tecnología de virtualización) para manejar tipos específicos de cargas de trabajo —desde procesamiento en lotes (“batch”) hasta aumento de servidor/almacenamiento durante las cargas pico.





## **Bibliografía**

- [1] Jordi Torres Viñals. (2011). Empresas en la nube. Ventajas y retos del cloud computing. Barcelona: Libros de cabecera S.L..
- [2] Jonathan Rasmusson. (2010). The Agile Samurai. Dallas, Texas: The Pragmatic Bookshelf.
- [3] MongoDB Architecture. (2015, Octubre 05). Recuperado de <https://www.mongodb.com/mongodb-architecture>.
- [4] Semantic UI User Interface is the language of the web. (2015, Octubre 05). Recuperado de <http://semantic-ui.com/>.
- [5] Meteor Documentation. (2015, Octubre 05). Recuperado de <http://docs.meteor.com/>.
- [6] JavaScript. (2015, Octubre 05). Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>.

