[CAPITULO I - INTRODUCCIÓN 1](#_Toc443516718)

[1.1 Antecedentes 1](#_Toc443516719)

[1.2 Análisis del Problema 1](#_Toc443516720)

[1.2.1 Definición del Problema 1](#_Toc443516721)

[1.3 Objetivo del Proyecto 2](#_Toc443516722)

[1.3.1 Objetivo General 2](#_Toc443516723)

[1.3.2 Objetivos Específicos 2](#_Toc443516724)

[1.4 Áreas Involucradas 2](#_Toc443516725)

[1.5 Justificación 2](#_Toc443516726)

[1.6 Alcance 3](#_Toc443516727)

[CAPITULO II – Sistema web 4](#_Toc443516728)

[2.1 Sistema web 4](#_Toc443516729)

[2.1.1 Sistemas web y su diferencia con las aplicaciones de escritorio 5](#_Toc443516730)

[2.1.2 Ventajas 5](#_Toc443516731)

[2.1.3 Desventajas 6](#_Toc443516732)

[2.2 Arquitecturas Cliente/Servidor 6](#_Toc443516733)

[2.2.1 Separación de funciones 7](#_Toc443516734)

[2.2.2 Modelos de distribución en aplicaciones cliente / servidor 7](#_Toc443516735)

[2.2.2.1 Presentación distribuida 7](#_Toc443516736)

[2.2.2.2 Aplicación distribuida 7](#_Toc443516737)

[2.2.2.3 Datos distribuidos 7](#_Toc443516738)

[2.2.3 Arquitecturas de dos y tres niveles 8](#_Toc443516739)

[2.3 Elementos de un sistema web 8](#_Toc443516740)

[2.3.1 El cliente 8](#_Toc443516741)

[2.3.2 El Servidor 8](#_Toc443516742)

[2.3.3 Protocolo HTTP 8](#_Toc443516743)

[2.4 Transferencia de páginas web 9](#_Toc443516744)

[2.5 Entornos web 10](#_Toc443516745)

[2.5.1 Internet 10](#_Toc443516746)

[2.5.2 Intranet 10](#_Toc443516747)

[2.5.3 Extranet 10](#_Toc443516748)

[2.6 HTML Hypertext Transfer Protocol 10](#_Toc443516749)

[2.7 Cloud Computing 10](#_Toc443516750)

[2.7.1 Historia 10](#_Toc443516751)

[2.7.2 Características 11](#_Toc443516752)

[2.7.3 Arquitectura de la nube 11](#_Toc443516753)

[2.7.3.1 Software como servicio (SaaS) 11](#_Toc443516754)

[2.7.3.2 Plataforma como Servicio (PaaS) 12](#_Toc443516755)

[2.7.3.3 Infraestructura como Servicio (IaaS) 12](#_Toc443516756)

[2.7.4 Ventajas 13](#_Toc443516757)

[2.7.5 Desventajas 13](#_Toc443516758)

[2.8 Software como Servicio (SaaS) 14](#_Toc443516759)

[2.8.1 Características 14](#_Toc443516760)

[2.8.2 Aplicaciones de una sola página (SPA) 14](#_Toc443516761)

[2.8.2.1 Estructura de un SPA 15](#_Toc443516762)

[2.8.2.2 Comunicación con el Servidor 16](#_Toc443516763)

[2.8.2.3 Transferencia de Estado Representacional (REST) API 16](#_Toc443516764)

[2.8.3 Diseño web Adaptable 17](#_Toc443516765)

[CAPITULO III - TECNOLOGIAS 18](#_Toc443516766)

[3.1 Plataforma de desarrollo 18](#_Toc443516767)

[3.1.1 Frameworks 18](#_Toc443516768)

[3.1.1.1 Angular JS 18](#_Toc443516769)

[3.1.1.2 CodeIgniter REST-Server 18](#_Toc443516770)

[3.1.2 Lenguajes de Programación 18](#_Toc443516771)

[3.1.2.1 JavaScript 18](#_Toc443516772)

[3.1.2.2 Características 19](#_Toc443516773)

[3.1.2.3 Justificación 19](#_Toc443516774)

[CAPITULO Iv – areas de aplicacion 20](#_Toc443516775)

[4.1 Instalación deportiva 20](#_Toc443516776)

[4.1.1 Teminologia 21](#_Toc443516777)

[4.1.1.1 Espacio deportivo 21](#_Toc443516778)

[4.1.1.2 Complejo deportivo 21](#_Toc443516779)

[4.1.1.3 Espacios complementarios 21](#_Toc443516780)

[4.1.1.4 Servicios auxiliares 21](#_Toc443516781)

[4.1.1.5 Cancha 21](#_Toc443516782)

[4.1.2 Clasificacion 21](#_Toc443516783)

[4.1.2.1 Espacios deportivos convencionales 21](#_Toc443516784)

[4.1.2.2 Espacios deportivos singulares 22](#_Toc443516785)

[4.1.2.3 Áreas de actividad deportiva 22](#_Toc443516786)

[4.2 Administracion de Horarios y Reservas 22](#_Toc443516787)

[4.2.1 Administracion de Horarios 22](#_Toc443516788)

[4.2.2 Reservas 25](#_Toc443516789)

[4.2.1.1 Reservas regulares 25](#_Toc443516790)

[4.2.1.2 Reservas provenientes de eventos 27](#_Toc443516791)

[4.2.1.2 Reservas procedentes de dias de mantenimiento 27](#_Toc443516792)

[Bibliografía 28](#_Toc443516793)

# 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 cuenten 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 contaran 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 – Sistema web

Este capítulo está dedicado a realizar una descripción completa de los elementos conceptuales que se manejan sobre sistemas web. Asimismo se detallan las características principales de cada elemento.

## Sistema web

Un sistema web o también denominado aplicación web según Sergio Lujan Mora es: “Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HyperText Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones (Figura 2.1).”

Estos tipos de aplicaciones son populares debido a lo práctico del navegador web, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea, etc. que son ejemplos bastante conocidos de aplicaciones web.

Una aplicación web contiene elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la aplicación responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

En vista de que algunas veces existen confusiones sobre páginas web estáticas y dinámicas, a continuación se hace una diferencia entre ellas.

Una **página web estática** tiene las siguientes propiedades:

* Realizadas principalmente en XHTML o HTML.
* Para modificación de contenidos es imprescindible acceder al servidor donde está alojada la página.
* El proceso de actualización es lento y manual.
* No ofrece bases de datos, interacción, etc.
* El usuario no puede seleccionar, ordenar o modificar los contenidos o el diseño de la página.
* Su diseño y desarrollo es más rápido y económico que el de una web dinámica.

Una **página web dinámica** tiene las siguientes propiedades:

* Muchas posibilidades en diseño y desarrollo.
* Utiliza varios lenguajes y técnicas de programación en su desarrollo.
* El proceso de actualización es muy sencillo y no requiere entrar en el servidor.
* Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico, etc.
* El usuario puede alterar el diseño, contenidos o presentación de la página a su gusto.
* Su diseño y desarrollo es más caro que el de una web estática, por su mayor complejidad y sus mayores prestaciones.

Por las características listadas anteriormente, se concluye que una aplicación web o sistema web es un tipo de página web dinámica, ya que concuerda con dichas características. Además, se considera que es un “sistema de información”, porque funciona como cualquier otra aplicación de escritorio, pudiendo almacenar grandes cantidades de información.

### Sistemas web y su diferencia con las aplicaciones de escritorio

En las aplicaciones de escritorio cada cliente debe instalar por separado el software. El cliente realiza peticiones a otro programa (el servidor) que le da respuesta. Una mejora en el servidor, como parte de la aplicación, requiere normalmente una mejora de los clientes instalados en cada computadora personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

A diferencia de lo anterior, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, como HTML que es soportado por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de extensiones tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario.

Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

### 2.1.2 Ventajas

A continuación, se muestra una lista de las ventajas que ofrecen este tipo de sistemas de información.

* **Ahorra tiempo:** se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
* **No hay problemas de compatibilidad**: basta tener un navegador actualizado para poder utilizarlas.
* **No ocupan espacio en nuestro disco duro.**
* **Actualizaciones inmediatas**: como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
* **Consumo de recursos bajo**: dado que toda (o gran parte) de la aplicación no se encuentra en nuestra computadora, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otra computadora.
* **Multiplataforma:** se pueden usar desde cualquier sistema operativo porque solamente es necesario tener un navegador.
* **Portables:** es independiente de la computadora donde se utilice (PC de sobremesa, portátil) porque se accede a través de una página web (solamente es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
* **La disponibilidad** suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
* **Los virus no dañan los datos** porque están guardados en el servidor de la aplicación.
* **Colaboración:** gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones en línea de calendarios u oficina.
* **Los navegadores ofrecen cada vez más y mejores funcionalidades** para crear "aplicaciones web enriquecidas" (Rich Internet application o RIA).

### 2.1.3 Desventajas

* **Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio**. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo.
* **La disponibilidad depende de un tercero**. El proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

### Arquitecturas Cliente/Servidor

Un sistema web, tradicionalmente funciona bajo una arquitectura Cliente / Servidor, ya que se despliega en una red de computadoras. Según Sergio Lujan Mora, esta arquitectura se define de la siguiente manera.

“Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Normalmente, los servidores son ordenadores potentes dedicados a gestionar unidades de disco (servidor de ficheros), impresoras (servidor de impresoras), tráfico de red (servidor de red), datos

(Servidor de bases de datos) o incluso aplicaciones (servidor de aplicaciones), mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores”.

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos.



*Figura numero(elaboración propia o dirección url??)… : Arquitectura Cliente / Servidor*

### Separación de funciones

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles.

* **Lógica de presentación.** La presentación de los datos es una función independiente del resto.
* **Lógica de negocio (o aplicación).** Los flujos de trabajo pueden cambiarse según las necesidades existentes de un procesador a otro.
* **Lógica de datos.** La gestión de los datos debe ser independiente para poder ser distribuida según las necesidades de la empresa en cada momento.

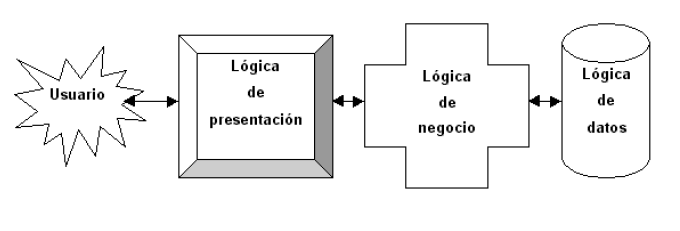


Figura N.(extraido del libro de lujan). : Separacion de funciones en arquitectura cliente servidor.

### 2.2.2 Modelos de distribución en aplicaciones cliente / servidor

Según como se distribuyan las tres funciones básicas de una aplicación (presentación, negocio y datos) entre el cliente y el servidor, podemos contemplar tres modelos: presentación distribuida, aplicación distribuida, datos distribuidos.

#### 2.2.2.1 Presentación distribuida

El cliente sólo mantiene la presentación, el resto de la aplicación se ejecuta remotamente. La presentación distribuida, en su forma más simple, es una interfaz gráfica de usuario, a la que se le pueden acoplar controles de validación de datos, para evitar la validación de los mismos en el servidor.

#### 2.2.2.2 Aplicación distribuida

Es el modelo que proporciona máxima flexibilidad, puesto que permite tanto a servidor como a cliente mantener la lógica de negocio realizando cada uno las funciones que le sean más propias, bien por organización, o bien por mejora en el rendimiento del sistema.

#### 2.2.2.3 Datos distribuidos

Los datos son los que se distribuyen, por lo que la lógica de datos es lo que queda separada del resto de la aplicación. Se puede dar de dos formas: ficheros distribuidos o bases de datos distribuidas.

### 2.2.3 Arquitecturas de dos y tres niveles

Una arquitectura de dos niveles está basada en un sistema gestor de bases de datos donde el cliente mantiene la lógica de la presentación, negocio, y de acceso a los datos, y el servidor únicamente gestiona los datos. Suelen ser aplicaciones cerradas que supeditan la lógica de los procesos cliente al gestor

de base de datos que se está usando.

En las arquitecturas de tres niveles, la lógica de presentación, la lógica de negocio y la lógica de datos están separadas, de tal forma que mientras la lógica de presentación se ejecutará normalmente en la estación cliente, la lógica de negocio y la de datos pueden estar repartidas entre distintos procesadores.

El objetivo de aumentar el número de niveles en una aplicación distribuida es lograr una mayor independencia entre un nivel y otro, lo que facilita la portabilidad en entornos heterogéneos.

### Elementos de un sistema web

Los elementos más importantes de una aplicación web son: El cliente, el servidor y el protocolo http.

### El cliente

El cliente web es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HyperText Markup Language (HTML) que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en Java. También se suelen emplear plug-ins3 que permiten visualizar otros contenidos multimedia (como Flash4), aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de incompatibilidad entre distintas plataformas. Por tanto, la misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (imágenes, sonidos, etc.).

### El Servidor

El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. En los sistemas Unix suele ser un demonio y en los sistemas Microsoft Windows un servicio.

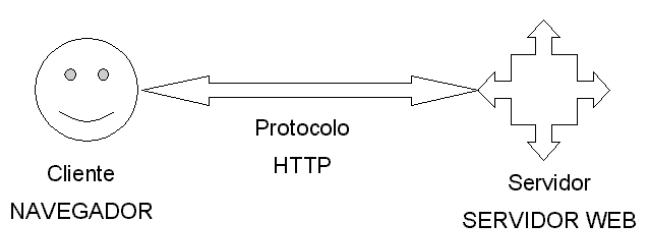
La parte servidor de las aplicaciones web está formada por páginas estáticas que siempre muestran el mismo contenido y por programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas dinámicas.

La programación en la parte del servidor se la realiza mediante tecnologías como ASP, ColdFusion, Java Server Pages (JSP), servlets, PHP, Javascript, etc. Todas ellas son muy similares, se basan en los mismos principios y ofrecen resultados equivalentes.

### Protocolo HTTP

El protocolo HTTP(Hypertext Transfer Protocol) Protocolo de transferencia de hipertexto, forma parte de la familia de protocolos de comunicaciones Transmission Control Protocol/Internet Protocol (TCP/IP), que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores.

En la figura siguiente se puede observar la relación entre los tres elementos de un sistema web.



*Figura N: Relación entre los elementos de un sistema web*

## Transferencia de páginas web

El proceso completo, desde que el usuario solicita una página, hasta que el cliente web (navegador) se la muestra con el formato apropiado, es el siguiente:

* El usuario especifica en el cliente web la dirección de la página que desea consultar: el usuario escribe en el navegador la dirección (Universal Resource Locator (URL)) de la página que desea visitar.
* El cliente establece una conexión con el servidor web.
* El cliente solicita la página o el objeto deseado.
* El servidor envía dicha página u objeto (o, si no existe, devuelve un código de error).
* Si se trata de una página HTML, el cliente inicia sus labores de interpretación de los códigos HTML. Si el cliente web encuentra instrucciones que hacen referencia a otros objetos que se tienen que mostrar con la página (imágenes, sonidos, animaciones multimedia, etc.), establece automáticamente comunicación con el servidor web para solicitar dichos objetos.
* Se cierra la conexión entre el cliente y el servidor.
* Se muestra la página al usuario.

Obsérvese que siempre se libera la conexión, por lo que ésta sólo tiene la duración correspondiente a la transmisión de la página solicitada. Esto se hace así para no desperdiciar innecesariamente el ancho de banda de la red mientras el usuario lee la página recibida.

Cuando el usuario activa un enlace de la página, se establece una nueva conexión para recibir otra página o elemento multimedia. Por ello, el usuario tiene la sensación de que está disfrutando de una conexión permanente cuando realmente no es así.

Un detalle importante es que para casa objeto que se transfiere por la red se realiza una conexión independiente. Por ejemplo, si el cliente web solicita una página que contiene dos imágenes integradas, se realizan tres conexiones: una para el documento HTML y dos para los archivos de las imágenes.

## Entornos web

Los sistemas web se emplean en tres entornos informáticos muy similares que suelen confundirse entre sí: Internet, intranet y extranet.

### Internet

Internet es una red global que conecta millones de ordenadores por todo el mundo. Su nacimiento se sitúa en 1969, en un proyecto de investigación del Departamento de Defensa de Estados Unidos. En 1998, la Internet tenía más de 100 millones de usuarios en todo el mundo, en diciembre de 2000 unos 400 millones y el número sigue creciendo rápidamente. Más de 100 países están conectados a este nuevo medio para intercambiar todo tipo de información.

### Intranet

Una intranet es una red de ordenadores basada en los protocolos que gobiernan Internet (TCP/IP) que pertenece a una organización y que es accesible únicamente por los miembros de la organización, empleados u otras personas con autorización.

Una intranet puede estar o no conectada a Internet. Un sitio web en una intranet es y actúa como cualquier otro sitio web, pero los cortafuegos (firewall) lo protegen de accesos no autorizados.

### Extranet

Una extranet es una intranet a la que pueden acceder parcialmente personas autorizadas ajenas a la organización o empresa propietaria de la intranet.

Mientras que una intranet reside detrás de un cortafuego y sólo es accesible por las personas que forman parte de la organización propietaria de la intranet, una extranet proporciona diferentes niveles de acceso a personas que se encuentran en el exterior de la organización.

## HTML Hypertext Transfer Protocol

HTML es un lenguaje de marcas (etiquetas) que se emplea para dar formato a los documentos que se quieren publicar en la WWW. Los navegadores pueden interpretar las etiquetas y muestran los documentos con el formato deseado.

Las páginas web o páginas HTML son unos cheros escritos en el lenguaje HTML. El desarrollo de estas páginas abarca un amplio grupo de tecnologías, desde las páginas más sencillas que sólo usan el lenguaje HTML hasta las más complejas que usan Dynamic HTML (DHTML), JavaScript, applets realizados en Java o componentes ActiveX.

## Cloud Computing

Según Oscar Ávila Mejía, cloud computing o computacion en la nube es : “Cuando se hace referencia a la nube, se está aludiendo a un termino con algunos años de historia y que es una forma metafórica de nombrar a Internet. Básicamente la computación en la nube consiste en los servicios ofrecidos a través de la red tales como correo electrónico, almacenamiento, uso de aplicaciones, etc., los cuales son normalmente accesibles mediante un navegador web”.

### Historia

El desarrollo de la computación en la nube comenzó a través de grandes empresas de servicios de

Internet como Google y Amazon los cuales construyeron su propia infraestructura. A partir de allí surgió una arquitectura: un sistema de recursos distribuidos de manera horizontal, introducidos como servicios virtuales de tecnologías de información (TI) escalados masivamente y manejados como recursos agrupados y configurados continuamente.

Aunque la implementación es reciente, la idea no es nueva ya que se ha discutido en el medio desde hace algunos años con distintos nombres tales como: “utility computing”, computación en demanda, computación elástica, o “grid computing” (no confundir con el procesamiento en red mencionado anteriormente). Haciendo una comparación de ideas y tecnologías entre las décadas de 1960 y 1970 con la época actual se tiene lo siguiente:

**Década 1960**

* Uso de “terminales tontas”2 que dependían de un sistema central más potente.
* La información se guardaba en el servidor.
* Se necesitaba conexión constante con el sistema central para funcionar correctamente.
* Imposibilidad de instalar aplicaciones.

**Siglo XXI**

* Terminales poco potentes pero autosuficientes(Netbooks, tablets, smartphones)
* La información se aloja en los servidores del proveedor de servicios aunque hay posibilidad de guardar información en la terminal del usuario.
* Se necesita conexión constante con el sistema central para hacer uso de todos los recursos.
* Dependiendo de la terminal, es posible instalar aplicaciones, aunque la idea es ejecutarlas a través de Internet.

### Características

* No es necesario disponer de un equipo potente, tan solo de un aparato con conexión a internet; esto debido a que el dispositivo del usuario no realizara ningún proceso complejo y los ficheros puede guardarse en la nube. Los servidores en donde se hallan los programas que se utilicen son los encargados de las tareas complicadas que antes se realizaba localmente.
* Con el uso del Cloud Computing no hay necesidad por parte del usuario de conocer la infraestructura detrás de esta, ya que pasa a ser una abstracción, “una nube” donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y con pocas fallas.
* Es auto reparable: En caso de surgir un fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de ´esta se genera uno nuevo.
* Posee un alto nivel de seguridad: El sistema esta creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad; de esto se ocupa el sistema proveedor que se encarga de cifrar los datos.
* Disponibilidad de la información: No se hace necesario guardar los documentos editados por el usuario en su computadora o en medios físicos propios ya que la información radicara en Internet permitiendo su acceso desde cualquier dispositivo conectado a la red (con autorización requerida).

### Arquitectura de la nube

La computación en la nube basa su arquitectura haciendo una separación entre hardware, plataforma y aplicaciones quedando las siguientes capas:

#### 2.7.3.1 Software como servicio (SaaS)

Se encuentra en la capa más alta y consiste en la entrega de aplicaciones completas como un servicio.

El proveedor de tecnologías de información y comunicación (TIC) ofrece el SaaS (Software as a Service).

Para ello dispone de una aplicación que se encarga de operar y mantener y que frecuentemente es desarrollada por ´el mismo. Con ella se encarga de dar servicio a multitud de clientes a través de la red, sin que estos tengan que instalar ningún software adicional.

Un ejemplo claro es la aplicación para el manejo del correo electrónico por medio de un navegador de Internet.

#### 2.7.3.2 Plataforma como Servicio (PaaS)

En orden descendente, PaaS (Platform as a Service) es la siguiente capa. Básicamente su objetivo se centra en un modelo en el que se proporciona un servicio de plataforma con todo lo necesario para dar soporte al ciclo de planteamiento, desarrollo y puesta en marcha de aplicaciones y servicios web a través de la misma. El proveedor es el encargado de escalar los recursos en caso de que la aplicación lo requiera, de que la plataforma tenga un rendimiento óptimo, de la seguridad de acceso, etc. Para desarrollar software se necesitan bases de datos, herramientas de desarrollo y en ocasiones servidores y redes.

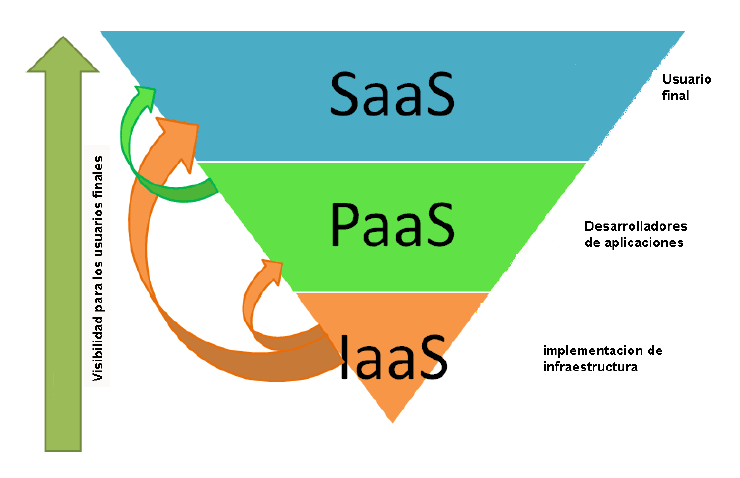
Con PaaS el cliente únicamente se enfoca en desarrollar, depurar y probar ya que la herramienta necesaria para el desarrollo de software es ofrecido a través de Internet, lo que teóricamente permite aumentar la productividad de los equipos de desarrollo.

Con el uso de PaaS, se abstrae del hardware físico al cliente, lo cual es interesante para muchos desarrolladores web, y es probable que llegue a reemplazar a las empresas de alojamiento tradicionales.

#### 2.7.3.3 Infraestructura como Servicio (IaaS)

IaaS (Infrastructure as a Service) corresponde a la capa más baja. La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, ruteadores, swtiches así como tiempo de computo evitando de esta manera tener un servidor local y toda la infraestructura necesaria para la conectividad y mantenimiento dentro de una organizaron. Con una IaaS lo que se tiene es una solución en la que se paga por consumo de recursos solamente usados: espacio en disco utilizado, tiempo de CPU, espacio para base de datos, transferencia de datos, etc.

Las IaaS permiten desplazar al proveedor la mayor parte de los factores relacionados con la gestión de las maquinas con el ahorro de costos al pagar solo por lo consumido y olvidarse de tratar con máquinas y su mantenimiento. Por otro lado, IaaS puede permitir una escalabilidad automática o semiautomática, de forma que se puedan contratar más recursos según los se requieran5.



*Figura N : Capas de computación en la nube*

### 2.7.4 Ventajas

* **Prestación de servicios a nivel mundial.** Las infraestructuras de cloud computing proporcionan mayor capacidad de adaptación, recuperación completa de pérdida de datos (con copias de seguridad) y reducción al mínimo de los tiempos de inactividad.
* **Una infraestructura 100% de cloud computing** permite también al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de software, ya que éste es provisto por el proveedor de la infraestructura o la plataforma en la nube. Un gran beneficio del cloud computing es la simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.
* **Implementación más rápida y con menos riesgos**, ya que se comienza a trabajar más rápido y no es necesaria una gran inversión. Las aplicaciones del cloud computing suelen estar disponibles en cuestión de días u horas en lugar de semanas o meses, incluso con un nivel considerable de personalización o integración.
* **Actualizaciones automáticas** que no afectan negativamente a los recursos de TI. Al actualizar a la última versión de las aplicaciones, el usuario se ve obligado a dedicar tiempo y recursos para volver a personalizar e integrar la aplicación. Con el cloud computing no hay que decidir entre actualizar y conservar el trabajo, dado que esas personalizaciones e integraciones se conservan automáticamente durante la actualización.
* **Contribuye al uso eficiente de la energía.** En este caso, a la energía requerida para el funcionamiento de la infraestructura. En los datacenters tradicionales, los servidores consumen mucha más energía de la requerida realmente. En cambio, en las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio.

### Desventajas

* **La disponibilidad de las aplicaciones** está sujeta a la disponibilidad de acceso a Internet.
* **Los datos "sensibles" del negocio no residen en las instalaciones de las empresas**, lo que podría generar un contexto de alta vulnerabilidad para la sustracción o robo de información.
* **La confiabilidad de los servicios depende de la "salud" tecnológica y financiera** de los proveedores de servicios en nube. Empresas emergentes o alianzas entre empresas podrían crear un ambiente propicio para el monopolio y el crecimiento exagerado en los servicios.
* **La disponibilidad de servicios altamente especializados** podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
* **La madurez funcional de las aplicaciones** hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.
* **Seguridad.** La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que éstos requieren.

## Software como Servicio (SaaS)

Software como Servicio (del inglés: Software as a Service, SaaS) es un modelo de distribución de software, en donde la compañía de tecnologías de información (IT) provee el servicio de mantenimiento, operación diaria y soporte del software usado por el cliente. Regularmente el software puede ser consultado en cualquier computador, esté presente en la empresa o no. Se deduce que la información, el procesamiento, los insumos y los resultados de la lógica de negocio del software está hospedado en la compañía de IT.

Las actividades son gestionadas desde alguna ubicación central, en lugar de hacerlo desde la sede de cada cliente, permitiendo a los clientes el acceso remoto a las aplicaciones mediante la web.

### 2.8.1 Características

Las características del software como servicio incluyen:

* Acceso y administración a través de una red.
* Actividades gestionadas desde ubicaciones centrales, en lugar de la sede de cada cliente, permitiéndoles tener acceso remoto a las aplicaciones a través de la web.
* La distribución de la aplicación es más cercana al modelo uno a muchos (una instancia con múltiples usuarios) que al modelo uno a uno, incluyendo arquitectura, precios, colaboración, y administración.
* Actualizaciones centralizadas, lo cual elimina la necesidad de descargar parches por parte de los usuarios finales.
* Multiplataforma, en donde el cliente puede ejecutar la aplicación desde diferentes sistemas operativos, incluso, diferentes tipos de dispositivos, tales como celulares, tabletas o computadores.

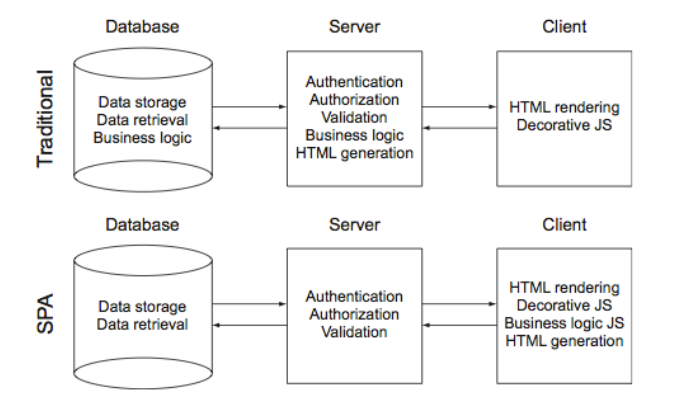
### 2.8.2 Aplicaciones de una sola página (SPA)

Una de las formas de implementar un servicio Saas (Software as a service), es desarrollando un SPA (Single page Application) en español, aplicación de una sola página, lo cual es una solución óptima para los servicios Saas.

Según Hung Ho Ngoc una aplicación de una sola página es: “Una aplicación de una sola página (SPA) es una aplicación web instalado en una sola página y hace que no vuelva a cargar la página durante su uso para proporcionar una mejor experiencia de usuario y aumentar el rendimiento”.

### 2.8.2.1 Estructura de un SPA

Cuando la aplicación web, se traslada al navegador, los requisitos para el rendimiento en el servidor son bastante importantes. La siguiente figura muestra cómo las SPA solucionan un problema de rendimiento con la lógica de negocio y el procesamiento de plantillas migran desde el servidor y el cliente.



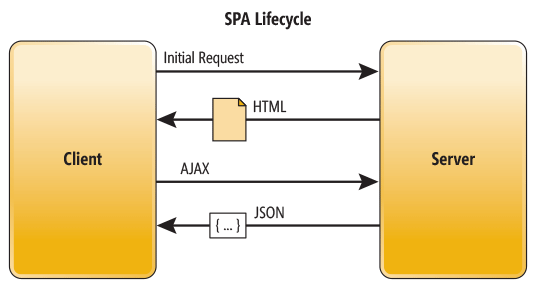
*Figura Nro : Responsabilidades de la base de datos, servidor y cliente en una aplicación web tradicional y una aplicación de una sola página (SPA)*

Como se puede ver en el gráfico, con una SPA, toda la lógica de negocio de la base de datos y el servidor (Tradicional) se desplazan al lado del cliente (SPA) con el fin de mejorar el rendimiento de la Base de datos y servidor. Esto porque una de las principales características de un servicio SaaS es que puede tener múltiples usuarios.

Una aplicación web tradicional, como se puede ver en la figura, realiza trabajos que hacen que las solicitudes tomen mucho mas tiempo para ser resueltas, lo cual se traduce en una experiencia baja de usuario, sobretodo tratándose de un servicio SaaS.

La forma en que una SPA funciona es básicamente de la siguiente manera: cuando el usuario interactúa con la página realizando alguna acción (pulsar un botón, enlaces u otros eventos) se realiza una petición asíncrona (AJAX) al servidor, cuya respuesta (generalmente XML, JSON o similar) es interpretada por la aplicación que actualizará el DOM (Document Object Model) **“sin recargar el sitio”**, permitiendo visualizar los cambios con alta velocidad y sin desperdicio de ancho de banda. Las SPA delegan muchas operaciones que anteriormente se realizaban en el servidor al lado del cliente, reduciendo así la complejidad del sistema.

La figura siguiente muestra de una forma clara el proceso de funcionamiento de una SPA, realizando peticiones asíncronas mediante tecnología AJAX.



*Figura N : Funcionamiento de un SPA, realizando peticiones asíncronas*

#### 2.8.2.2 Comunicación con el Servidor

Las peticiones que realiza una SPA, normalmente resultan en datos crudos (XML o JSON), o son plantillas HTML nuevos que están siendo regresados. En los casos donde el HTML regresa como resultado, JavaScript en el cliente actualiza el área parcial de DOM (Document Object Model). Cuando el dato crudo es regresado, el JavaScript de lado del cliente procesa el XML o el JSON para hacer la traducción de los datos crudos a un HTML la cual es usada para actualizar la información del área parcial del DOM.

Con todo lo descrito anteriormente sobre aplicaciones de una sola página (SPA), se concluye que la forma de comunicación entre el cliente (SPA) y el servidor, es mediante peticiones asíncronas, las cuales requieren únicamente de la URL del recurso solicitado. Razon por la cual en el lado del servidor se hace uso de los llamados “Servicios REST”.

#### 2.8.2.3 Transferencia de Estado Representacional (REST) API

Según Hung Ho Ngoc un servicio REST es : “Transferencia de estado representacional (REST) es la arquitectura sin estado, basado en el protocolo HTTP para representar el modelo de cómo debería funcionar los sistemas web modernos. Los servicios REST utilizan la tecnología y protocolos de web existente para acceder y manipular los recursos que utilizan los verbos (métodos de solicitud HTTP).”

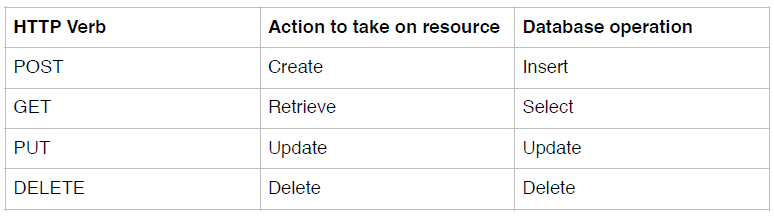


Tabla N : Cómo las acciones de HTTP son relevantes para el funcionamiento de bases de datos.

Hay una serie de verbos para el enfoque REST como GET, POST, PUT y DELETE para manipular los datos en la base de datos. Estos verbos corresponden a un CRUD tradicional (Crear, Leer, actualizar, eliminar) las operaciones en una base de datos. Además, REST también tiene 5 categorías de respuestas: información general, una petición correcta, redirigir, error en el cliente y el error en el lado del servidor. La respuesta depende del tipo de petición HTTP, como se indica en la siguiente tabla:

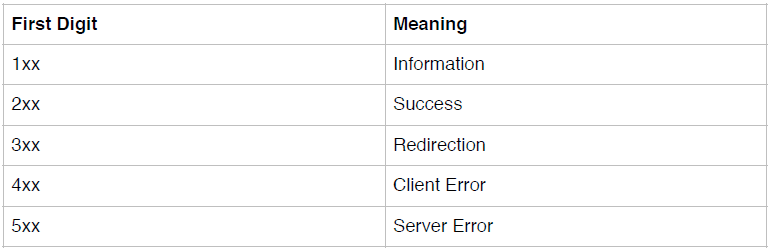


Tabla N : Errores y respuestas comunes en un servicio REST.

### 2.8.3 Diseño web Adaptable

# CAPITULO III - TECNOLOGIAS

Este capítulo está dedicado a realizar una descripción completa de las herramientas tecnológicas, que se han de utilizar para el desarrollo de un sistema web, con características de una aplicación de una sola página (SPA) como se había descrito en el capítulo II.

## 3.1 Plataforma de desarrollo

La siguiente plataforma de desarrollo, ha sido elegida tomando en cuenta las características de un servicio SaaS (Software como Servicio) y una aplicación de una sola página (SPA).

### 3.1.1 Frameworks

El uso de un framework ayuda con la seguridad, reducción de tiempo de desarrollo de una aplicación. Además nos permite tener código optimizado, ordenado y entendible. Por tanto se ha hecho la elección de los siguientes frameworks: Angular Js, framework para JavaScript y CodeIgniter REST-Server, framework para PHP.

#### 3.1.1.1 Angular JS

La razón principal por la que se ha elegido a Angular Js como framework de Javascript, es que ha sido desarrollado específicamente para aplicaciones web de una sola página (SPA).

Según la página oficial, Angular se define de la siguiente manera: “AngularJS es un marco estructural para aplicaciones web dinámicas. Permite utilizar HTML como lenguaje de plantillas y permite extender la sintaxis HTML para expresar los componentes de una aplicación clara y concisa. El enlace de datos de AngularJS y la inyección de dependencia elimina gran parte del código que de otro modo tendría que escribir. Y todo sucede dentro del navegador, lo que lo convierte en un socio ideal con cualquier tecnología de servidor.”

Angular js es un framework javascript de codigo abierto, que implementa el patrón MVC (Model-View-Controller o Model-View-ViewModel). Ha sido desarrollado por Google.

#### 3.1.1.2 CodeIgniter REST-Server

### 3.1.2 Lenguajes de Programación

Los lenguajes de programación que se hacen uso para el desarrollo del sistema web son: PHP y JavaScript, PHP del lado del servidor y JavaScript del lado del cliente.

#### 3.1.2.1 JavaScript

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.

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]

#### 3.1.2.2 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.

#### 3.1.2.3 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.
* Existe una gran cantidad de bibliotecas para trabajar con JavaScript.

# CAPITULO Iv – areas de aplicacion

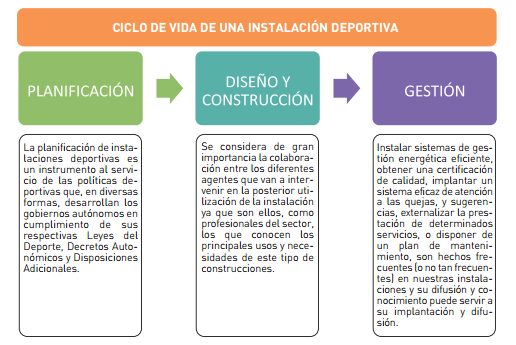
En este capitulo se realiza una definicion sobre complejos deportivos, sus caracteristicas y tipos. Se hace una diferencia entre instalacion deportiva y complejo deportivo.Se definen conceptos secundarios sobre complejos deportivos, como instalacion deportiva, cancha, servicios auxiliares, etc.Se hace una descripcion completa de los procesos de registro de campos deportivos, realizacion de una reserva, etc, asi como el diagrama de flujo de proceso para cada una de ellas.

## 4.1 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 instalacion deportiva*

*Fuente:*

### 4.1.1 Teminologia

#### 4.1.1.1 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.

#### 4.1.1.2 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.

#### 4.1.1.3 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.

#### 4.1.1.4 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.

#### 4.1.1.5 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 : futbol-sala, futbol, bascket, etc.

### 4.1.2 Clasificacion

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

#### 4.1.2.1 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 m² 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 m². 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.

#### 4.1.2.2 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.

#### 4.1.2.3 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.

## 4.2 Administracion de Horarios y Reservas

### 4.2.1 Administracion de Horarios

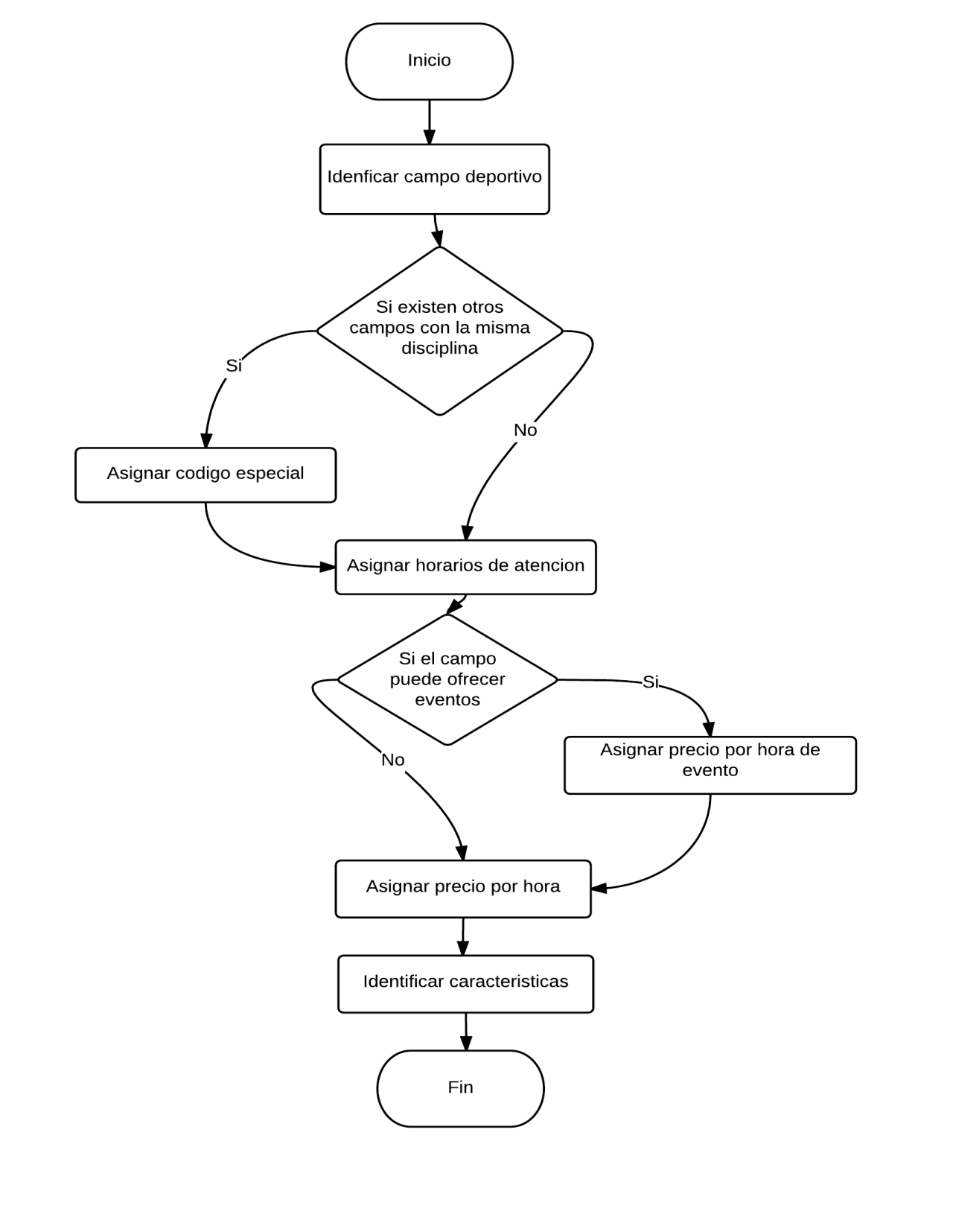
Administracion de horarios es el proceso de desarrollo, mantenimiento y comunicación de los horarios de tiempo y de recursos.

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 codigo, ya que pueden existir varios campos con la misma disciplina deportiva, asimismo identifica las caracteristicas del campo, como pueden ser: tipo de suelo, el precio por cada hora de reserva. Una de las caracteristicas mas importantes es el de la hora de atencion del campo deportivo, el cual sera definido por el administrador del complejo al momento de hacer el registro del campo.

La division de horarios que se realiza por dia, es decir, el numero de veces maximo por dia 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 diagram de flujo del proceso se puede observer en la imagen inferior.



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

*Fuente : Elaboracion propia*

### 4.2.2 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:

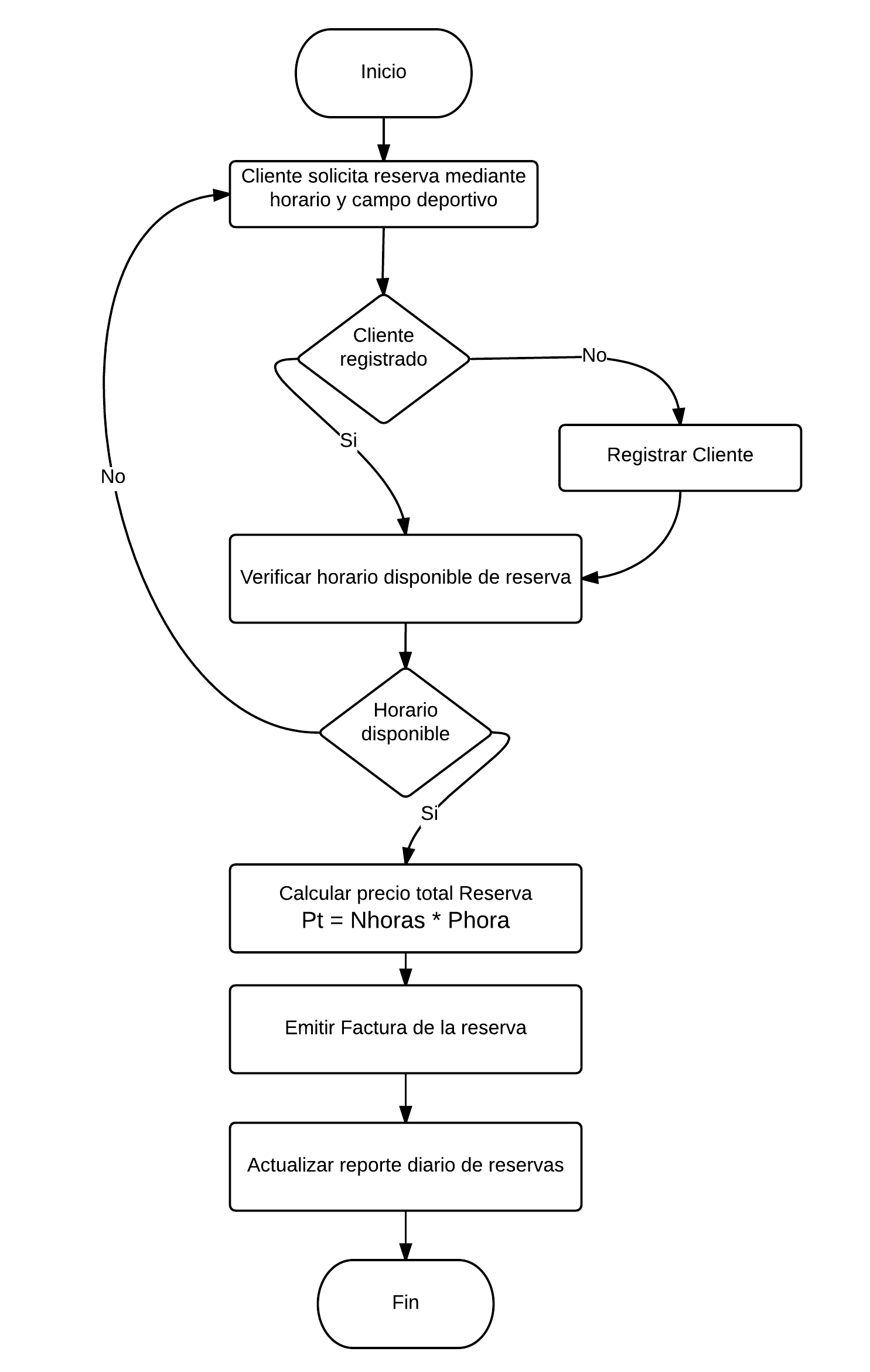
* Reservas regulares.
* Reservas provenientes de eventos.
* Reservas procedentes de dias de mantenimiento.

#### 4.2.1.1 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 peticion de uso de algun 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 mas veces han usado el campo deportivo, formando una cola de clientes prioritarios. Asi, si no existe una prereserva por algun 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, deacuerdo al numero de horas de la reserva, se calcula el precio total de la reserva, deacuerdo a la siguiente formula Pt = Nhoras \* Phora, esto sirve para el reporte diario de reservas y para la emision de facturas

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

**

*Figura 8 : Diagrama de flujo para una reserva*

*Fuente : Elaboracion propia*

#### 4.2.1.2 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. Porteriormente el encargado de reservas del complejo deportivo notifica al administrador que dicho campo es reservado para un evento(el cual tendra un precio superior al de una reserve regular). El administrador prepara un documento, el cual representa un contrato para la reserva del evento, que entre los datos mas importantes de dicho contrato estan: el nombre del administrador, nombre del cliente con sus repectivas 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)

#### 4.2.1.2 Reservas procedentes de dias de mantenimiento

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

La caracteristica principal de este tipo de reserva es que no genera ningun ingreso economico para el complejo, mas al contrario, es una perdida economica pero necesaria para mantener en buen estado el campo deportivo.

Esta reserva la autoriza el administrador del complejo deportivo.

# 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.

[6] JavaScript. (2015, Octubre 05). Recuperado de <https://developer.mozilla.org/es/docs/Web/JavaScript>.