



**SISTEMA DE GESTIÓN DE TURNOS EN LÍNEA:
ANÁLISIS, DISEÑO Y DESARROLLO DE UNA SOLUCIÓN CONFIGURABLE**

**TRABAJO FINAL DE GRADUACIÓN
LICENCIATURA EN INFORMÁTICA
UNIVERSIDAD EMPRESARIAL SIGLO 21**

2009

**BORTAGARAY, MARTÍN
INF 415**

ÍNDICE

1 - Introducción.....	4
1.1 - Internet y Comercio electrónico	4
2 - Antecedentes del Negocio.....	7
3 - Fundamentación	8
3.1 - Clientes de nuestros Clientes.....	9
4 - Objetivo	10
5 - Alcance	11
5.1 - Límites	12
5.2 - Producto Potencial	12
6 - Marco Teórico.....	13
6.1 - Marco de Antecedentes	13
6.2 - Análisis del Proceso de Negocio	17
6.2.1 - Actores	18
6.2.2 - Canales.....	19
6.3 - Metodología de Análisis.....	19
6.3.1- Análisis y Diseño Práctico para Sistemas Cliente/Servidor con GUI	19
6.4 - Tecnologías de Desarrollo	22
6.4.1 - Consideraciones Iniciales	22
6.4.2 - Cuándo utilizar formularios Windows Forms.....	23
6.4.3 - Cuándo utilizar formularios Web Forms	24
6.4.4 - Nuestra Solución: Aplicación Web.....	24
6.4.5 - Estructura de las páginas Web	25
6.4.6 - Tecnologías utilizadas desde el lado del Servidor.....	26
6.4.6.1 - ASP.NET.....	29
6.4.6.2 - Web Services.....	34
6.4.7 Tecnologías utilizadas desde el lado del Cliente	36
6.4.7.1 JavaScript.....	36
6.4.7.2 AJAX	37
6.4.8 - Base de Datos	39
6.4.8.1 - SQL Server 2005	40
7 - Propuesta.....	41
8 - Estimación y Presupuesto.....	44
9 - Análisis	46
9.1 - Plan General	46
9.2 - Modelo de Contexto	48
9.3 - Modelo de Eventos	50
9.3.1 - Lista de Eventos	50
9.3.2 - Diccionario de Eventos	51
CU - Registrar Nueva Entidad	52
CU - Registrar Operador Administrador de Entidad	52
CU - Iniciar Sesión	53

CU – Registro Inicial de Espacios	54
CU – Gestión de Espacios por Entidad	55
CU – Gestión de Horarios por Espacio.....	56
CU – Gestión de Días no Laborables	57
CU – Gestión de Operadores	59
CU – Visualización de Agenda de Espacios	60
CU – Bandeja de Entrada de Autorización de Turnos	61
CU – Carga de turnos por otros canales	62
CU – Modificar/Eliminar turnos registrados.....	63
CU – Gestión de Clientes	64
CU – Registrar Nuevo Usuario	65
CU – Inicio de Sesión de Usuario	66
CU – Consultar Disponibilidad de Turnos.....	67
CU – Solicitar Turno	68
CU – Administrar turnos solicitados.....	69
CU – Configurar Datos Propios	69
9.4 - Prototipo de Interfaz	71
Pantalla Inicial.....	71
Pantalla de Inicio de Sesión	72
Pantalla de Registro de Nueva Entidad	72
Pantalla de Gestión de Turnos	73
Pantalla de Autorización de Turnos.....	74
Pantalla de Configuración (Espacios)	75
9.5 - Modelo de Información	76
9.5.1 - Detalle de Diagrama de Entidad Relación	77
Bibliografía	78
Páginas Web Consultadas	78
Anexo 1: Referencia de Honorarios	79
Anexo 2: Estándares de Codificación	80

1 - INTRODUCCIÓN

Hoy en día las empresas y negocios necesitan agilizar y automatizar sus procesos, el factor **tiempo** se ha vuelto una variable esencial en la ecuación final de utilidades y es por esto que los empresarios deben administrarlo de la forma más eficiente posible. El tiempo no se puede ahorrar, el tiempo que pasó y no se aprovechó se perdió para siempre, no puede recuperarse, es un recurso no renovable y no puede comprarse.

Por esto, las herramientas que permitan administrar con eficacia el tiempo se vuelven fundamentales para el éxito de cualquier negocio. El objetivo de este trabajo no apunta a "cualquier negocio", apunta a negocios o empresas que "venden" tiempo, es decir que su fin último y productivo es vender la mayor cantidad de tiempo posible teniendo en cuenta la disponibilidad de este recurso escaso. Contar con una herramienta que permita optimizar la administración y gestión del tiempo se vuelve vital para este tipo de negocios.

El siguiente Trabajo Final de Graduación correspondiente al Seminario Final de la Carrera de Licenciatura en Informática de la Universidad Empresarial Siglo 21 tiene como fin el diseño y presentación de esta herramienta, un sistema informático web para la gestión de reservas y turnos online.

La solución que se plantea es genérica, es decir que la finalidad es que pueda utilizarse en cualquier empresa o negocio que administre espacios por tiempo, es decir o bien alquile espacios y/u objetos o preste servicios, en ambos casos se necesita administrar tiempo y este sistema tiene esa finalidad. Llamaremos de aquí en más a este tipo de negocios y/o empresas como **clientes potenciales**.

Si bien la solución es genérica, se toman los requerimientos de un negocio en particular para poder relevar adecuadamente todas las características del proceso de reservación y partiendo de este punto poder hacer extensivo los requerimientos para todos los negocios y/o empresas que potencialmente puedan usar esta herramienta. Este negocio es un club deportivo que ofrece canchas de pádel, squash básquet y vóley en alquiler.

1.1 - Internet y Comercio electrónico

Es de público conocimiento que el auge y el uso de Internet tiene un crecimiento exponencial en los últimos años tanto en nuestro país como en el mundo.

El acceso a Internet se está volviendo cada día más veloz, económico y conveniente debido a dispositivos y formas de conexión existentes. Cada vez es mayor el número de consumidores que cuando necesita algo, un producto, un servicio o cualquier información, recurre, como primera fuente, a Internet porque allí podrá informarse, comparar, evaluar alternativas de manera rápida, sencilla y anónima hasta adquirir lo necesario. Pero no podemos detenernos aquí, porque de esta manera convertiríamos a la web simplemente en un medio publicitario más y si bien no deja de serlo tiene muchas otras ventajas y características distintivas por sobre sus competidores (televisión, publicidad gráfica, y otros). Internet es un medio de comunicación masivo, cuya audiencia está en constante crecimiento, además de ser una audiencia cualificada y de perfil comercial, sin embargo, tiene una característica que lo convierte en único; la posibilidad de interacción con su receptor. Así hoy por hoy es muy común realizar trámites, compras y ventas vía Internet, y, si bien en nuestro país aun existe cierto temor a ingresar datos personales críticos como el número de tarjeta de crédito, esta modalidad crece día a día. Pero no sólo las ventas utilizan el poder de interacción de Internet, podemos encontrar muchas más aplicaciones, herramientas para todo tipo de negocios, empresas o personas que permiten colaborar en las tareas diarias y de esta manera optimizar los procesos. Podemos por ejemplo encontrar una listado de aplicaciones web en el siguiente sitio: <http://www.webramientas.com>.



Veamos algunos números de este fenómeno: Según los datos publicados por el Instituto Nacional de Estadísticas y Censos (INDEC) de Argentina, los accesos a Internet desde Marzo del 2007, al mismo mes de 2008, habrían aumentado un 14,6% en el sector residencial, y un 11% en el de empresas. Este incremento permite que Argentina ahora posea un total de casi 3 millones de usuarios residenciales, y 272 mil usuarios de empresas, frente a los 2.9 millones y 271



mil registrados, respectivamente, durante el mes de diciembre pasado.

En cuanto a los servicios de banda ancha, estos totalizan a marzo pasado, unos 2.2 millones de usuarios, lo cual es una buena porción del total de usuarios con acceso a la red.

Pero no sólo creció el mercado de conexión a la red. El comercio electrónico en los últimos dos años, registró un crecimiento acumulado del 163%, hasta alcanzar los US\$ 739 millones anuales. El estudio destaca que sólo el 5,2% de la población argentina compra por Internet. En Capital Federal y Gran Buenos Aires 21 por ciento de los usuarios de Internet realizaron compras en la Web, y en el interior esa porción baja al 14 por ciento, una cifra que ofrece proyecciones interesantes si se tiene en cuenta que hasta ahora en las provincias esta modalidad era casi nula. Argentina, México, Brasil y Chile son los principales impulsores del comercio electrónico en la región. Estos países concentran el 80 por ciento de las operaciones y se proyecta para el período 2008-2010 un crecimiento anual de 40 por ciento, según una investigación de Visa Internacional.

"El primer paso ya se ha dado. Todas las empresas se han dado cuenta de la importancia de estar en la red. Paralelamente, la confianza en el sistema de compra virtual va aumentando. El 60 por ciento de los usuarios busca bienes o servicios, pero recién se anima a comprar en la web a los dos años de antigüedad como usuario", destacó Pablo Tedesco, director asociado de Prince & Cooke.

Por esta razón las empresas deberían ver en la web oportunidades de crecimiento en el mercado, nos enfocaremos en las pequeñas y medianas empresas (PYMES) ya que en las grandes corporaciones por lo general Internet ya es una herramienta adoptada al punto tal de ser uno de los pilares del funcionamiento. Las empresas que no adopten Internet en su operatoria diaria estarán condenadas al estancamiento y a la pérdida de su participación en el mercado. La satisfacción del cliente y adelantarse a sus necesidades es la misión por decirlo de alguna manera, genérica de estas empresas y si Internet puede colaborar a este fin, bienvenido sea.

2 – ANTECEDENTES DEL NEGOCIO

El cliente que se tomó como referencia para obtener los requerimientos de la herramienta es un club deportivo, su nombre es TANCAT Deportes y tiene dos sucursales en la Ciudad de Córdoba, su negocio radica principalmente en el alquiler de canchas de Pádel, Básquet, Squash, Vóley y Tenis de Mesa, también tiene servicio de bar, parrilla y venta de indumentaria deportiva. El negocio es netamente familiar y recién en los últimos años con la apertura de la nueva sucursal se contrataron empleados. Todos los locales poseen PC con conexión a Internet que sólo es utilizada como entretenimiento del empleado de turno o eventualmente para atender consultas vía mensajería instantánea. Las reservas son tomadas en su mayoría telefónicamente, derivando los clientes a que llamen a las otras sucursales en caso de no tener disponibilidad, o bien de manera presencial.

El negocio no utiliza sistemas administrativos informatizados, las reservas y la ocupación de las canchas son llevadas en una planilla diaria de papel, donde se anota también el consumo de los clientes durante su estadía en el local para luego realizar el ticket final mediante una controladora fiscal. No se utiliza ningún sistema informático de facturación y/o control de stock.

Resulta para el negocio un trabajo tedioso y repetitivo tener que comunicarle al cliente telefónicamente la disponibilidad de horarios y no contar con la información del estado de ocupación de las otras sucursales en tiempo real y es en este problema donde se buscará una oportunidad de desarrollo de una herramienta para llegar a una solución óptima.

Se realizó hasta aquí una breve descripción del proceso de negocio de este cliente en particular, y si bien este trabajo plantea el desarrollo de una solución genérica es muy importante relevar y conocer el funcionamiento del negocio de este tipo de empresas ya que, todas presentan similares características.

Resolviendo la problemática de este negocio es donde este trabajo encontrará una oportunidad de distinción. Se podrán satisfacer los requerimientos relevados y se encuentra una oportunidad para brindar una solución integral a los clientes potenciales.

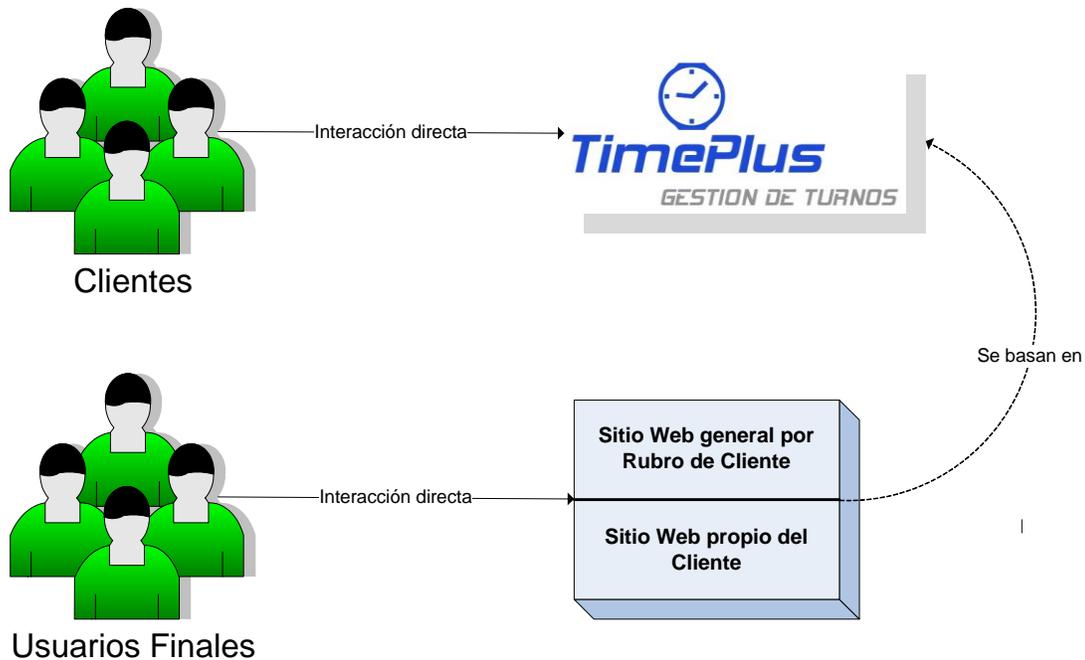
Ahora bien, ¿Qué sucede desde el lado de los usuarios o consumidores finales?, es decir con los clientes de nuestros clientes potenciales. Supongamos por un momento que somos un consumidor final y que queremos jugar al pádel,

necesitaríamos reservar una cancha para practicar este deporte, por lo tanto lo natural sería llamar por teléfono a los clubes conocidos y preguntar por la disponibilidad hasta dar, si tenemos suerte, con el objetivo. ¿No sería interesante tener la información de todas las canchas de la ciudad (o sector que se determine) para así poder elegir con comodidad y sin pérdida de tiempo? Mediante internet se puede brindar una herramienta que nos permita esto y mucho más, como por ejemplo poder realizar la reserva directamente a través de la web, sin tener que llamar por teléfono o dirigirnos al lugar físico propiamente dicho. El ejemplo se centra en el alquiler de un espacio deportivo, pero podría aplicarse a cualquier rubro de similares características, esto es, un espacio físico o servicio que quiera ser contratado y necesite de una reserva previa.

3 - FUNDAMENTACIÓN

Los negocios necesitan disponer de su información en tiempo y forma y de manera consistente y segura y, lo que es más importante, necesitan brindarle a sus clientes un servicio cada vez más veloz y eficiente. Conjugando estas dos premisas podemos encontrar una buena oportunidad para crear una herramienta que ayude al cumplimiento de los objetivos planteados y mejore sustancialmente los procesos de reservación. Este trabajo apunta a resolver el problema presentado, desde el punto de vista de los usuarios de volver más dinámico y menos engorroso el proceso de reserva de un espacio y desde el punto de vista del negocio a organizar, dinamizar y centralizar la toma de reservas, además de brindar a los clientes un nuevo canal de comunicación (Internet) que, como vimos, está en franco avance, con un sin límite de oportunidades.

Según lo explayado en el párrafo anterior se puede inferir que la solución puede separarse en dos partes, una destinada a nuestros clientes potenciales, como dijimos llamaremos **clientes potenciales** a los negocios, empresas, organizaciones o personas que utilicen nuestra aplicación para gestionar y administrar sus agendas de turnos o reservas; y otra, destinada a los usuarios finales, llamaremos **usuarios finales** a los *clientes de nuestros clientes*, son los consumidores, los pacientes o toda organización o persona que desee reservar un turno en algunos de nuestros "clientes potenciales". A continuación vemos una figura con lo recientemente expuesto:



3.1 - Clientes de nuestros Clientes

Cuando una persona (o porque no, una organización) necesita solicitar un turno o realizar una reserva de algún espacio o servicio, la forma común de actuar es llamar telefónicamente y solicitarlo, esto si ya conoce de antemano el lugar donde quiere hacer dicha reserva, si esto no es así deberá realizar antes un proceso previo de adquisición de información y evaluación de alternativas, por diferentes medios que estén a su alcance, contactos personales, publicidades, directorio telefónico, entre otros. Y si cuando damos con el lugar indicado no hay disponibilidad para reservar lo que queremos, debemos iniciar nuevamente el proceso, hasta poder así (si es que antes no nos dimos por vencidos) lograr nuestro objetivo. Puede ser en algunos casos tedioso y complicado para los clientes.

¿No sería interesante disponer de una herramienta que resuma todos estos pasos en un mismo lugar? El desafío que plantea el trabajo es diseñar y desarrollar esta herramienta y que su grado de usabilidad sea óptimo.

La herramienta a la que hacemos mención es un sitio Web, donde los usuarios (consumidores) ante la necesidad de solicitar un turno puedan conectarse, consultar, evaluar alternativas, ya que estarían disponibles todas las diferentes opciones, y reservar, vía Internet. La tarea no es fácil primeramente porque para que sea un éxito deberíamos instalar en la mente de los consumidores la idea de que cuando necesiten

hacer una reserva ingresen a nuestro sitio, y segundo porque necesitamos que todos los negocios relacionados sean nuestros aliados y estén dispuestos a publicar sus disponibilidades.

4 - OBJETIVO

Diseñar y desarrollar un sistema de gestión de reservas y turnos en línea, a través de Internet, que brinde a los clientes (negocios aliados) una herramienta de gestión y administración centralizada y eficiente de sus agendas de turnos de sus espacios de alquiler y que, ofrezca a los usuarios finales la posibilidad de buscar, seleccionar y reservar turnos disponibles en tiempo real, de estos espacios.

Definiciones con relación al objetivo de trabajo

- Clientes:** Negocios aliados, son las empresas, negocios, organizaciones o personas que contratan el servicio de esta aplicación y además de administrar su agenda de turnos, ponen a disposición su disponibilidad de horarios para la reserva de sus espacios.
- Espacios:** Llamamos "Espacio" a todo lugar u objeto, físico, que se destine a alquiler o sea utilizado para prestar un servicio.
- Agenda:** Cada espacio de alquiler tiene su agenda, la cual se divide por lo general en períodos (días) y fracciones (horas) de tiempo y donde van a consignarse las reservas correspondientes.
- Usuarios Finales:** Son los clientes de nuestros clientes, son los deportistas (canchas de pádel, squash, tenis, entre otras), son los pacientes (profesionales, instituciones médicas, entre otros), son los consumidores (solárium, services, entre otros) y toda aquella persona u organización que desee consultar y/o reservar un turnos en la agenda de algún espacio de nuestros clientes.

5 - ALCANCE

La esencia de este trabajo es que pueda abarcar todos los rubros de productos o servicios que requieran de alguna manera una reservación previa para ser consumidos, como por ejemplo restaurantes, cines, teatros, espectáculos, Profesionales que prestan sus servicios por medio de turnos (médicos, odontólogos, psicólogos, fisioterapeutas, entre otros), Servicios de reparación, gimnasios, piscinas, solárium, por nombrar los ejemplos más relevantes. Pero a los fines de este trabajo de tesis y para acotar los contenidos vamos a enfocarnos en un solo rubro, este es, **espacios deportivos de alquiler**, que es el rubro del cliente de donde obtuvimos los requerimientos generales. Pero bien, la solución está diseñada para que en el futuro puedan implementarse en negocios de otros rubros como los antes mencionados.

Como vimos, se puede dividir el trabajo en dos partes interactuantes cuyo producto final serán **dos** sitios o aplicaciones web diferentes en relación al usuario final, así cuando los usuarios finales sean los consumidores:

Con respecto a los negocios que ofrecen sus espacios:

- ✓ Sitio propio de administración del sistema
 - Configuración de cantidad y tipos de espacios y sus características: frecuencia de agenda, días y horarios de atención, prestadores asociados.
 - Carga de reservas ingresadas por otros canales (presencial, telefónico, mensajería instantánea o mail)
 - Administración de agenda por espacio de alquiler
- ✓ Validación de turnos solicitados online
- ✓ Manejo de validaciones propias o genéricas.
- ✓ Calificación de usuarios clientes
- ✓ Presentación de datos históricos por diferentes filtros y exportación de datos.

Con respecto a los usuarios finales (consumidores) el sistema contemplará:

- ✓ Activación de cuenta de usuario con datos comprobables básicos (email y teléfono) sin costo alguno.
- ✓ Sección de actualización de perfil (datos de usuario)

- ✓ Búsquedas avanzadas de espacios¹ por Deporte, Zona (Provincia, Localidad, Barrio), Día, Hora, Precio y Disponibilidad.
- ✓ Consultas vía email a un negocio en particular.
- ✓ Realizar la reserva de un espacio disponible online, obteniendo la confirmación en tiempo real o con demora predeterminada dependiendo de la modalidad de operación del negocio en particular.
- ✓ Las confirmaciones de reservas pueden configurarse para poder recibirse por distintos canales: Email, SMS e Ingreso al portal.
- ✓ Que pueda registrarse en su perfil un historial de comportamientos en anteriores reservas. Esto es, luego de realizada la reserva y consumido (o no) el servicio/producto, el negocio calificará positiva o negativamente al usuario si es que cumplió o no con su reserva.

5.1 - Límites

- ✓ No se manejarán tarifas por espacio de alquiler por ser estas muy subjetivas y variables, sería muy engorroso su configuración por parte del usuario administrador. Se contará con una grilla para la registración de consumos.
- ✓ La aplicación no contempla, por el momento, el manejo de información financiera ni estadística para las entidades.

5.2 – Producto Potencial

Hasta aquí definimos los límites y alcances de la solución para el presente trabajo, pero podemos pensar en este sistema como un producto comercial potencial donde podríamos enriquecerlo con algunas funcionalidades extras para darle mayor valor agregado. Entre estas funcionalidades está pensado incluir.

- ✓ Soporte de posicionamiento georeferenciado
 - Con esta funcionalidad podemos brindarle al usuario final un mapa detallado de cómo llegar al lugar donde realizó la reserva desde su domicilio como así también incluir filtros de búsqueda de espacios según distancia.

¹ Denominaremos "espacios" a todo lugar físico que se ofrezca en alquiler.

- ✓ Pago de servicio mediante Tarjeta de Crédito
 - Permite que el usuario final pague el monto (o un porcentaje) del servicio por adelantado. Esto permite al negocio asegurar la reserva y disminuir el ausentismo.
- ✓ Módulo Financiero
 - Este módulo brindaría al negocio un soporte financiero donde podría llevarse la facturación y los gastos. No se pretende reemplazar programas contables pero es una buena manera de llevar el control asociado directamente a la venta de los servicios.
- ✓ Aplicación para dispositivos móviles
 - Actualmente el advenimiento de dispositivos móviles (celulares, palms, smartphones) con acceso a internet nos posibilita poder proyectar la solución para que pueda ser utilizada desde estos dispositivos.

6 - MARCO TEÓRICO

6.1 – Marco de Antecedentes

Se evaluarán soluciones o aplicaciones con características similares a este trabajo en el mercado, que puedan, de alguna manera, contribuir en algunos aspectos y significar de qué manera esta solución aportará características novedosas.

El análisis será de menor a mayor en relación al grado de compatibilidad de requerimientos, existen varios sitios que presentan características similares. Como primer ejemplo pueden nombrarse algunos sitios webs de negocios particulares que ofrecen dentro del mismo la posibilidad de reservar sus instalaciones o servicios online, pero por lo general sin ninguna inteligencia, sólo a modo de envío de consulta.

- Para un Video Club:
 - ✓ <<http://www.planetavideoclub.com.ar/reservas.php>>
(Sitio consultado el 26/03/2009)
- Para una clínica de tratamiento capilar:
 - ✓ <http://hairrecovery.com.ar/turnos/agenda_web_hair_lacapital.asp>
(Sitio consultado el 26/03/2009)

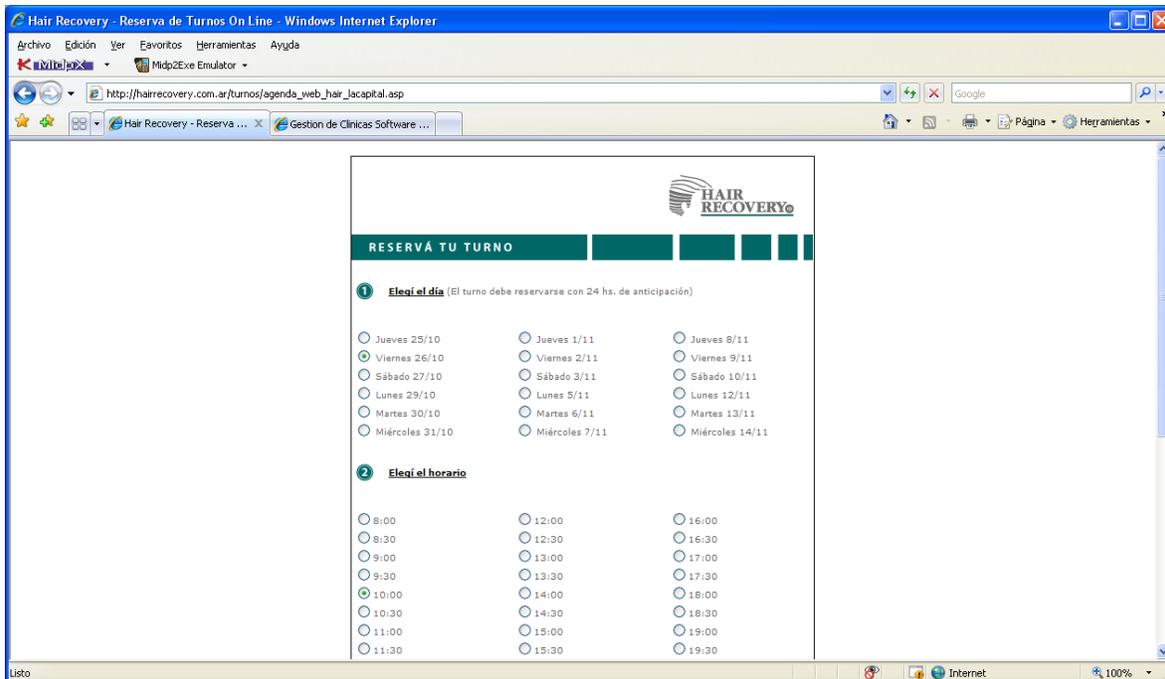


Figura 1. Solicitud de turnos online en HairRecovery.com.ar

Esta aplicación solicita el ingreso de Día, Hora, Sucursal y Datos personales y aclara que la asignación del turno solicitado está sujeta a disponibilidad, por lo tanto vemos que carece de inteligencia, es simplemente a modo de solicitud – contacto.

- Podemos ver otro ejemplo en la solicitud de turnos online del Sanatorio Allende (Córdoba):
 ✓ <<http://www.sanatorioallende.com/Turnos-Reservar-Paso1.asp>>
 (Sitio consultado el 28/03/2009)

Esta aplicación ya contiene cierta inteligencia, consta de varios pasos para solicitar el turno:

1. Identificación del usuario
2. Selección de Lugar de Atención
3. Selección de Especialidad Médica
4. Selección del Profesional
5. Selección de la fecha del turno. En este paso se muestran los días y horarios de atención del profesional.
6. Una vez seleccionada la fecha se muestran los turnos “disponibles” para este profesional.
7. Se selecciona el horario y se procede a reservar.

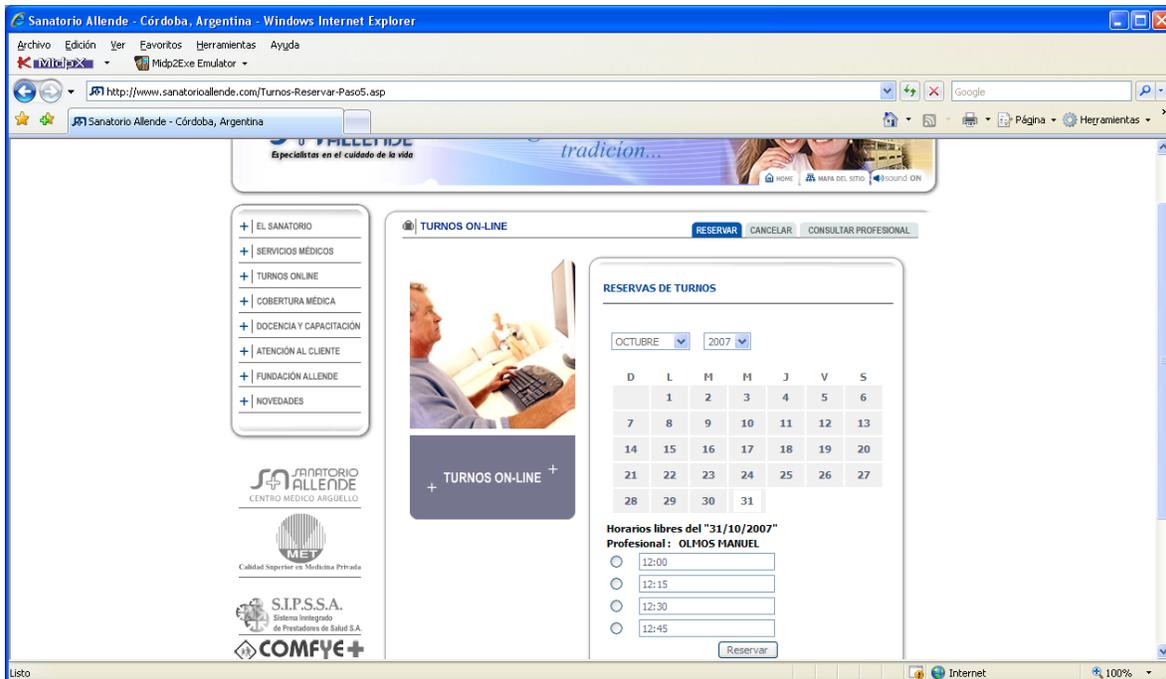


Figura 2. Solicitud de turnos online del Sanatorio Allende

- Por último veremos una aplicación que presenta características bastante similares a nuestra solución:

✓ <<http://www.iturnos.com/>>

Es una aplicación web para la reserva de turnos para diferentes rubros, es decir, genérica. Provee a consultorios, empresas o negocios el servicio de reserva de turnos on-line, a través del cual los clientes o pacientes autorizados ingresan a una agenda, visualizan los horarios disponibles y realizan la reserva del servicio ofrecido.

Este sistema tiene un costo de instalación de U\$S 18 por única vez y luego U\$S 18 mensuales por agenda.

Se puede resumir la operatoria de la aplicación en los siguientes pasos:

1. Carga de datos de la empresa – cliente, quien va a ofrecer y otorgar los turnos
2. Carga de datos del administrador, que es quien estará habilitado para configurar el entorno de operación. Tiene acceso a todas las funciones de la aplicación
3. Carga de datos de los eventuales operadores del sistema quienes tendrán funcionalidades limitadas

4. Carga de agendas. Se establecen los días, horarios y frecuencia de división de turnos como así también la configuración de las restricciones de operadores.
5. Carga de horarios laborales para la agenda especificada
6. Carga de bloqueos, se configura los días que no se trabaja, sea por feriados, fines de semana, enfermedad, u otros.
7. Pantalla de administración de agenda, donde se podrá cargar un nuevo turno, consultar los existentes, editarlos o cancelarlos. Todas estas acciones pueden ser informadas a los clientes (pacientes) por medio de email o sms.

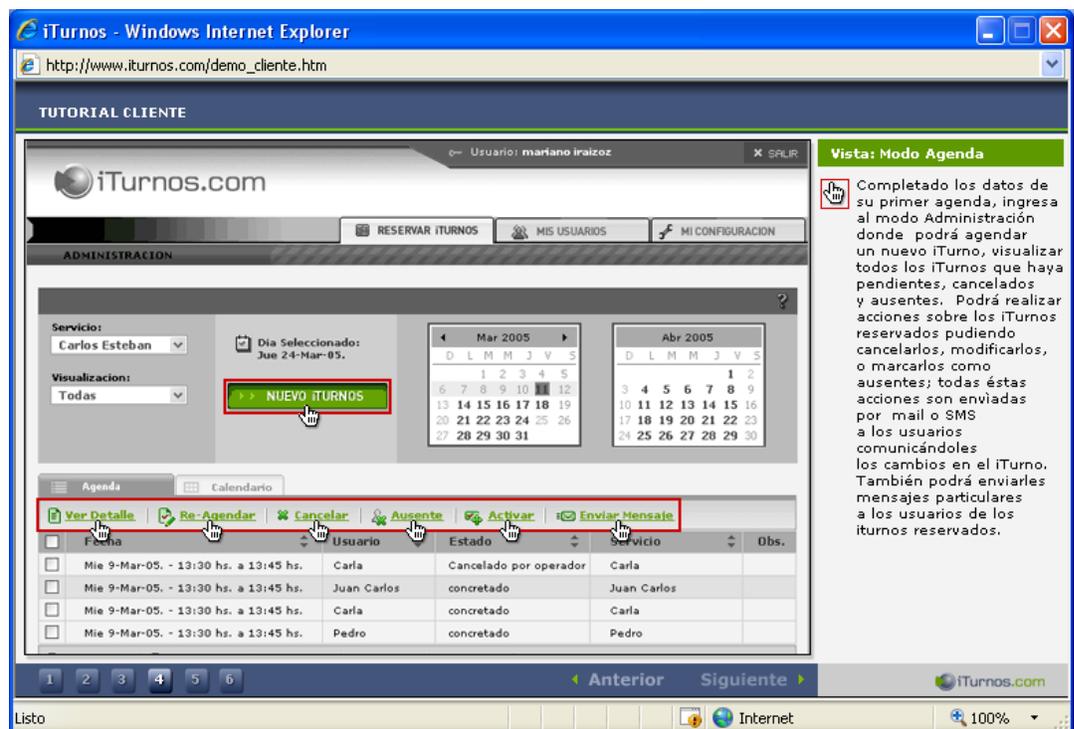


Figura 3. Pantalla de Modo Agenda de iTurnos.com

8. Modo calendario: Se visualiza la agenda como calendario, donde por medio de colores se podrá distinguir fácilmente los horarios disponibles y los ocupados.

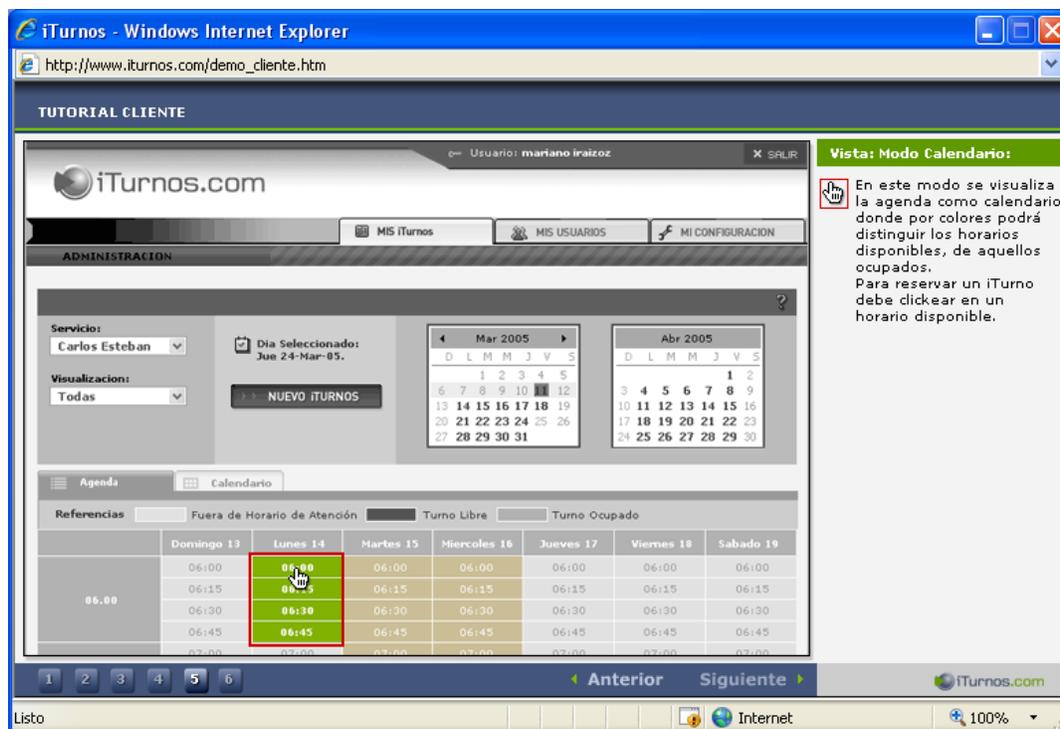


Figura 4. Pantalla de Modo Calendario de iTurnos.com

9. Vista de usuarios: En esta sección se pueden dar de alta o borrar clientes o pacientes.

La solución planteada en este trabajo comparte muchas características con este sitio, sobre todo para el Sitio Web de Administración, serían muy semejantes, **el valor agregado de nuestra solución con relación a iTurnos es el sitio Web para los consumidores finales que todos los negocios aliados lo que permitiría realizar búsquedas por diferentes criterios.**

iTurnos propone que cada negocio tenga su sitio web propio donde pueda adaptarse la aplicación.

6.2 – Análisis del Proceso de Negocio

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. Las entradas son requisitos que deben tenerse

antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes².

Si aplicamos esta definición al tipo de negocio objetivo de este trabajo final de graduación se puede ver que las entradas serían los clientes que necesitan del servicio de una empresa, la función aplicada es la prestación del servicio misma y la salida serían las utilidades. Tenemos que tener presente que nuestros clientes potenciales no venden bienes materiales, venden **tiempo**, por lo tanto es imprescindible optimizar su utilización, el tiempo que no se vendió está perdido, no puede recuperarse y tiene un elevado costo para el negocio.

6.2.1 - Actores

Según el relevamiento realizado en nuestro cliente puntual, podemos identificar los actores del proceso, estos son, en primera medida los "clientes del negocio", otro actor principal es el "negocio", es decir, por ejemplo "Tancat Deportes" en cada una de sus dos sucursales. Los demás actores son los empleados o recepcionistas de reservas.

Entonces podemos identificar los siguientes **actores** del sistema:

- **Cientes del negocio** que utiliza nuestra aplicación (Clientes de nuestros clientes). Algunos ejemplos:
 - Deportistas que desea alquilar una cancha para practicar algún deporte determinado.
 - Personas que deseen reservar un lugar para consumir algún servicio determinado.
 - Pacientes que deseen reservar un turno con un determinado profesional.
- **Negocio o Empresa** que utiliza nuestra aplicación (Nuestro Clientes). Algunos ejemplos:
 - Clubes deportivos que brinden canchas en alquiler (Pádel, tenis, squash, fútbol, vóley, básquet, entre otros)
 - Consultorios Médicos, Kinésicos, Odontológicos. Psicólogos.
 - Talleres de reparaciones.
 - Centros de Belleza, solárium.
 - Gimnasios.
- **Receptores de reservas**, son las personas que administran la agenda de turnos o reservas del negocio en cuestión.

² http://es.wikipedia.org/wiki/Proceso_de_negocio

6.2.2 - Canales

En nuestro cliente (Tancat Deportes), según lo relevado, las reservas son captadas por tres canales posibles identificados; por teléfono, personalmente o por mensajería instantánea por Internet. Por lo general en la mayoría de nuestros clientes potenciales los canales de recepción de reservas son los mencionados sobre todo el telefónico.

El proceso es como sigue, las reservas son tomadas por medio de alguno de estos tres canales y registradas en una planilla, en el caso de que el cliente no sea conocido (o no de la confianza necesaria) y que la reserva se tome por los canales telefónico o mensajería Web se solicita al cliente que se acerque a las instalaciones a fin de abonar la seña correspondiente en un plazo estipulado.

Las planillas de registración de reservas son diarias y constan de 7 columnas por espacio de alquiler:

- ✓ Hora (de 10:00 a.m. a 3:00 a.m., con intervalos de 30 minutos)
- ✓ Nombre (del cliente que realiza la reserva)
- ✓ L (si requiere de iluminación artificial)
- ✓ Seña
- ✓ Saldo (Valor del espacio por hora por cantidad de horas menos la seña)
- ✓ Total (Monto total a cobrar, aquí también se suman los consumos extras (bebidas, alquiler de insumos, entre otros, cuya cuenta es realizada en un cuaderno aparte)
- ✓ Observaciones

Estas planillas no tienen ningún soporte digital y son almacenadas en carpetas anuales.

6.3 - Metodología de Análisis

6.3.1- Análisis y Diseño Práctico para Sistemas Cliente/Servidor con GUI

A los fines de realizar el Análisis funcional del Sistema utilizaremos el Método estructural propuesto por **David A. Ruble**, se justifica su utilización en cinco premisas expuestas por el autor, Ruble establece que para que una metodología sea buena debe ser:

1. Motivar la actividad pretendida
2. Ser completa
3. Ser modificable
4. Producir productos contra los que se pueda medir el avance
5. Ser fácilmente aprovechable en la fase subsecuente

La metodología "Análisis y Diseño Práctico Para Sistemas Cliente / Servidor con GUI³" propuesta por **David A. Ruble** resalta la importancia de un buen análisis y diseño previos a la construcción del software.

Según Ruble el primer paso a dar en un proyecto de desarrollo de software, es el de la construcción de un *Plan General*. Dicho plan establece las *metas* y *objetivos* a cumplir por el proyecto, determina cuándo se ha logrado un objetivo y define responsabilidades. Ruble hace una distinción entre metas y objetivos:

- ✓ **Meta:** es el fin máximo a alcanzar por el proyecto, su razón de ser
- ✓ **Objetivos:** son las actividades específicas que se deben lograr para alcanzar la meta.

Ruble propone tres grandes modelos para el análisis:

- Modelo de Contexto.
- Modelo de Eventos.
- Modelo de Información.

Modelo de Contexto

Define el alcance del nuevo sistema como un **diagrama**. Donde el *círculo* representa al sistema propuesto, los *rectángulos* a las entidades que se comunicarán con el sistema y las *flechas* indican la dirección del flujo de la información. Dichas entidades son denominadas por Ruble como *Agentes Externos*. Un agente externo puede ser una persona, una organización, un cliente e incluso otros sistemas.

³ Interfaz Gráfica de Usuario o **Graphic User Interface** por sus siglas en inglés.

Modelo de Eventos

Define los *requerimientos* del sistema. Para ello dispone de dos herramientas: la lista de eventos y el diccionario de eventos:

- **Lista de Eventos:** cataloga cada evento que el sistema debería contemplar con una sintaxis sujeto-verbo-objeto
- **Diccionario de Eventos:** resulta más útil tanto para el analista como para el diseñador ya que detalla un evento en particular en

Estímulo → Actividad → Respuesta.

Modelo de Información

El propósito de este modelo es crear el Diagrama Entidad-Relación (**DER**), identificando entidades, atributos y relaciones. Información que será sumamente útil al momento de diseñar la base de datos. Ruble señala que sin importar la notación gráfica que se le de a las *cardinalidades* de las relaciones sólo existen **cuatro** combinaciones posibles:

- Cero a Uno
- Cero a Muchos
- Uno a Uno
- Uno a Muchos

La cardinalidad Muchos a Muchos no está permitida y debería resolverse con la creación de una entidad que permita dicha relación. A modo de ejemplo utilizaremos el símbolo asterisco (*) para Muchos y el número uno (**1**) para Uno. De este modo una relación Muchos a Muchos como ser Clientes – Productos podría resolverse como lo muestra la siguiente figura:



Luego estaríamos en condiciones de elaborar el Diccionario de Datos que determina las propiedades de cada atributo incluyendo: opcionalidad, tipo de dato, rango, unidad de medida, precisión y valores restringidos.

Aquí concluiría la fase de Análisis, luego Ruble propone los siguientes pasos para elaborar el diseño del sistema

- Modelo Arquitectónico: Mapea los requerimientos esenciales de la fase de análisis hacia una arquitectura tecnológica para determinar la configuración óptima.
- Diseño: es el proceso de determinar cuál de muchas posibles soluciones es la mejor para lograr lo que se necesita hacer.
 - Componentes internos
 - Nivel de Acceso a Datos
 - Diseño de Base de Datos
 - Capa de Acceso a Datos
 - Nivel de Negocio
 - Capa de Reglas de Negocio
 - Nivel de Aplicación
 - Diseño de Interfaz
 - Capa de Aplicación
 - Testeo
- Implementación: Etapa final en la cual el sistema es instalado y queda funcionando con la conformidad del usuario.

6.4 - Tecnologías de Desarrollo

6.4.1 - Consideraciones Iniciales

La primera decisión que se debe tomar en cuanto a que tecnologías de desarrollo utilizar en el armado del proyecto es si la interfaz de usuario será con formularios Windows Forms o formularios Web Forms, o sea si vamos a desarrollar una Aplicación Web o una Aplicación de Escritorio.

Ambos disponen de compatibilidad total en tiempo de diseño dentro del entorno de desarrollo y pueden proporcionar una interfaz de usuario completa y funciones de aplicación avanzadas para solucionar los problemas del negocio. Debido a esta paridad

de características, podría resultar difícil decidir qué tecnología es adecuada para una aplicación determinada.

Algunas consideraciones de la aplicación podrían hacer obvia la opción. Por ejemplo, si se está desarrollando un sitio Web accesible al público en Internet, lo ideal sería desarrollar la aplicación utilizando las páginas de formularios Web Forms. Si se está generando una aplicación muy receptiva y de alto rendimiento que necesite funcionalidad total del equipo de cliente, como una aplicación de productividad para la oficina, la mejor opción sería utilizar formularios Windows Forms. Sin embargo, en otros casos es posible que la elección no esté tan clara.

6.4.2 - Cuándo utilizar formularios Windows Forms

Se desarrollan aplicaciones con formularios Windows Forms cuando se desee que la aplicación cliente sea responsable de la mayoría de las tareas de procesamiento de una aplicación. Estas aplicaciones cliente incluyen aplicaciones de escritorio de Win32 desarrolladas tradicionalmente en versiones anteriores de Visual Basic y Visual C++. Ejemplo de ello son las aplicaciones gráficas o de diseño, los sistemas de entrada de datos, los sistemas de punto de venta y los juegos.

Estas aplicaciones se basan en la eficacia del equipo de escritorio al procesar y presentar el contenido de alto rendimiento. Algunas aplicaciones de Windows Forms pueden estar completamente auto contenidas y ejecutar todo el proceso de la aplicación en el equipo del usuario. A menudo, los juegos se escriben de este modo. Otras aplicaciones pueden formar parte de un sistema mayor y, fundamentalmente, utilizan el equipo de escritorio para procesar los datos proporcionados por el usuario. Por ejemplo, un sistema de punto de venta normalmente necesita una interfaz de usuario rápida y sofisticada que se crea en el equipo de escritorio, pero está enlazado a otros componentes que ejecutan el procesamiento en segundo plano.

Dado que una aplicación para Windows que utilice formularios Windows Forms se genera en un marco de trabajo Windows, tendrá acceso a los recursos del sistema del equipo cliente, que incluyen los archivos locales, el Registro de Windows, la impresora, etc. Se puede restringir este nivel de acceso para eliminar cualquier riesgo para la seguridad o problema potencial que pueda provocar un acceso no deseado. Además, los formularios Windows Forms pueden utilizar las clases gráficas GDI+ de .NET Framework para crear una interfaz gráficamente eficaz, que suele ser un requisito de las aplicaciones de extracción de datos o de juegos.

6.4.3 - Cuándo utilizar formularios Web Forms

Se utilizan formularios Web Forms de ASP.NET para crear aplicaciones que principalmente tienen una interfaz de usuario del explorador. Naturalmente, entre ellas se incluyen las aplicaciones que se ponen a disposición del público a través del World Wide Web como, por ejemplo, las aplicaciones de comercio electrónico. Sin embargo, puede utilizar formularios Web Forms no sólo para crear sitios Web; muchas otras aplicaciones se prestan a utilizar también una "interfaz fina", por ejemplo un manual del empleado basado en una intranet o una aplicación de beneficios. Una ventaja importante de cualquier aplicación de formularios Web Forms es que no tiene ningún costo de distribución. Los usuarios ya tienen instalada la única parte de la aplicación que necesitan, el explorador.

Las aplicaciones con formularios Web Forms son, por definición, independientes de la plataforma; es decir, son aplicaciones "de envergadura". Los usuarios pueden interactuar con la aplicación independientemente del tipo de explorador que tengan e, incluso, del tipo de equipo que utilicen. Al mismo tiempo, las aplicaciones de formularios Web Forms pueden optimizarse para aprovechar las características incorporadas a los exploradores más modernos, tales como Microsoft Internet Explorer 7.0, a fin de mejorar su rendimiento y velocidad de respuesta.

Los formularios Web Forms proporcionan algunas características útiles incluso en contextos que no son Web. Puesto que se basan en HTML, las aplicaciones Web Forms son las indicadas para aplicaciones que hacen un uso intensivo de recursos de texto del tipo que sean, sobre todo en las que el formato del texto es importante. Mientras que las aplicaciones basadas en explorador suelen tener un acceso limitado a los recursos del sistema de usuarios, esta limitación hace que las aplicaciones Web Forms sean útiles en casos en los que se desea evitar el acceso de los usuarios a partes de la aplicación.

6.4.4 - Nuestra Solución: Aplicación Web

La solución a desarrollar tendrá dos aplicaciones distintas, una a la que accederán los usuarios finales, es decir quienes necesiten realizar una reserva, a través de Internet y por medio de un browser o navegador, por lo que esta aplicación debe necesariamente realizarse como una página web, bajo la modalidad de formularios Web Forms.

La segunda aplicación es el Sistema de Administración para nuestros clientes, donde se realizarán las configuraciones y se receptaran y visualizarán las reservas realizadas por los usuarios, podríamos para esta aplicación evaluar la posibilidad de realizarla bajo Windows Forms, pero optamos por realizarlo también bajo la metodología de formularios Web Forms (Aplicación Web) para poder centralizar los datos y procesos de todos los clientes y poder brindar una solución configurable y dinámica. Por lo tanto definimos que:

Esta solución será desarrollada como una **Aplicación Web**, tanto el módulo Web (usuarios finales) como el Módulo Administrativo (clientes del negocio), esto es los usuarios y clientes accederán a un **Servidor Web** a través de Internet (o una Intranet) por medio de un **browser o navegador** (Internet Explorer, Mozilla Firefox, Safari, Google Chrome, por nombrar algunos). Un navegador de Internet interpreta documentos escritos en **HTML**, que previamente han obtenido de un servidor Web. Como tal en una Aplicación Web tenemos una parte del lado del cliente y otra del lado del servidor, es decir, cierta parte del código puede ejecutarse directamente en la PC del cliente por medio del navegador y otra parte va a ejecutarse del lado del servidor por medio de un lenguaje compilado.

Se realizará a continuación una breve introducción acerca de la estructura y composición de las Páginas Web como así también de las tecnologías utilizadas para el desarrollo de la solución tanto del lado del servidor como del lado del cliente.

6.4.5 - Estructura de las páginas Web

Se utilizará el lenguaje de estructuración XHTML acrónimo inglés de **eXtensible Hypertext Markup Language** (lenguaje extensible de marcado de hipertexto), ya que este es el pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, como pueden ser textos, hipervínculos o links e imágenes.

Para la presentación de las páginas se utilizarán hojas de estilo en cascada o CSS (acrónimo inglés de Cascade Style Sheet) como medio para definir la presentación (colores, tipos de letra, tamaños de letra) de los documentos estructurados con

XHTML. Estas permiten separar la estructura de un documento XHTML de su presentación. La información de estilo puede ser adjuntada a cada documento y así permitir que los cambios se puedan realizar en un solo lugar (archivo) y estos se reflejen en todo el sitio. Al igual que XHTML las especificaciones de las hojas de estilo son formuladas por el World Wide Web Consortium entonces se estarán respetando los estándares actuales de Internet.

6.4.6 - Tecnologías utilizadas desde el lado del Servidor

Se realizó una evaluación de las distintas plataformas posibles a utilizar para el desarrollo de la solución, comparando lo que son quizás, los tres lenguajes más utilizados en la actualidad: PHP, JSP y ASP .NET.

PHP

PHP es la tecnología Web más extendida en el momento. Nació para trabajar en Linux con servidor Apache, pero hoy en día puede alojarse en casi cualquier tipo de servidor. El código fuente está en abierto por lo que los bugs están muy controlados e inmediatamente solucionados, y además tiene una gran cantidad de módulos prefabricados que ya vienen instalados de fábrica en los servidores, y que no hay más que aprender a utilizar. Su sintaxis es muy similar a C, quizás algo más simple, y destaca que fue creada para programar páginas Web, aunque también se puede programar en local. Se comunica con bases de datos sin necesidad de usar ODBC. Las últimas versiones ya trabajan con orientación a objetivos. Hasta ahora es un lenguaje de script, no compilado.

JSP

Los Servlets de Java son muy comunes en aplicaciones Web potentes, como bancos o grandes empresas. Comparte muchas de las ventajas de ASP.NET, sobretodo en cuanto a la programación modular y orientada a objetos, pero sus carencias son también muy destacadas. Sobre todo su bajo enfoque de cara al usuario da mucho trabajo para presentar páginas Web al navegador. Para muy grandes aplicaciones suele elegirse JSP en lugar de PHP, dado que PHP es un lenguaje de más bajo nivel que JSP y dificulta la modularización y organización por capas de la aplicación.

ASP .NET

ASP.NET rompe totalmente con el pensamiento de script que se tenía hasta el momento. El cambio en la arquitectura es radical. De hecho, lo único que mantiene de ASP es el nombre, el propietario y la evolución de Visual Basic a Visual Basic .NET

(VB.NET)... el resto es todo nuevo. Dado que la Web no se lee secuencialmente sino que se compila, lo primero que llama la atención es el enorme incremento de velocidad de respuesta del servidor. Además, al compilarse, el incremento en seguridad y fortaleza es muy grande.

ASP.NET introduce el concepto del code-behind, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el de código. Con ello se facilita la programación de aplicaciones en múltiples capas, lo que en definitiva se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado. Por tanto, cualquier cambio de especificaciones minimiza los cambios en la aplicación y maximiza la facilidad de mantenimiento.

Asimismo, ASP.NET sirve tanto para Webs sencillas como para grandes aplicaciones. La orientación a objetos y la naturaleza compilada permiten hacer uso de herramientas de creación de Webs, las más importantes de la familia del Visual Studio, que facilitan mucho la tarea de programación. Estas herramientas permiten hacer Webs sencillas y de bajas prestaciones en un tiempo record, así como llevar el mantenimiento de grandes aplicaciones de forma más sencilla.

Resumiendo, tenemos mayor velocidad, mayor potencia, mayor seguridad, mayor facilidad de mantenimiento y herramientas de trabajo, pero las ventajas no paran aquí. A continuación se enumeran algunas otras que no tienen ASP, PHP o JSP:

- ✓ Caché: se puede almacenar en la caché del servidor tanto páginas enteras, como controles personalizados o simples variables. En páginas críticas con mucha carga de base de datos nos es muy útil almacenar datos de la base de datos en la caché, reduciendo enormemente el consumo de recursos.
- ✓ Carpetas especializadas, como por ejemplo `app_code` que compila automáticamente las clases que se alojan en él, o la carpeta `app_theme` que alojan ficheros que marcan los temas de estilos de la Web.
- ✓ Los archivos de configuración `Web.config` y `Machine.config` permiten realizar operación de configuración en ficheros que hasta ahora había que realizar en el servidor.
- ✓ La adaptación automática del código devuelto a los dispositivos que le acceden. Una misma página puede servirnos para el Internet Explorer, para el Pocket Internet Explorer desde una PDA o para un navegador de un móvil cualquiera.
- ✓ La eliminación total de la necesidad de frames con la introducción de las masterpages.

- ✓ La extraordinaria compatibilidad con XML y los servicios Web.
- ✓ La multitud de controles Web que permiten mucha funcionalidad con poco código. Desde enlace con las bases de datos o enseñar fácilmente todos los datos, hasta simples etiquetas, hiperenlaces o generadores de imágenes.
- ✓ Se puede utilizar hasta cuarenta lenguajes distintos para el desarrollo en ASP.NET, aunque en el 95% de las aplicaciones se usa C#, VB.NET o J#.

A continuación se presenta un cuadro comparativo con relación a las características de cada lenguaje:

	PHP	ASP .NET	JSP
Orientación a Objetos	3	4	4
Separación de Interfaz - Lógica	3	4	4
Productividad	3	4	3
Librería de Funciones	4	3	3
Debugging	2	4	3
Imágenes Dinámicas	Si	Si	Si
Herramientas provistas	Zend Studio	Visual Studio	Sun Java Studio Creator
Mantenimiento de estado de controles	No	ViewState	JavaBeans
Controles de Servidor	No	HTMLControls WebControls	JavaBeans
Seguridad en General	4	4	4
Manejo de Excepciones	Si	Si	Si
Sesiones	Cookie/Cookieles s	Cookie/Cookieless	Cookie
Autenticación Integrada	Ninguna	Windows Formularios Passport	Básica Formularios Digest Certificate
Modo de Configuración de la Aplicación	Php.ini	Jerarquía Archivos .config	Web .xml
<i>* Las Calificaciones son de 1 a 5</i>			

Figura 5. Cuadro comparativo Lenguajes Web

Según lo expuesto anteriormente la solución será desarrollada en Lenguaje ASP .NET, utilizando C#.net como lenguaje de codificación.

Para lograr aún mayor universalidad y entendimiento respecto al código fuente del Sistema a desarrollar se utilizarán estándares de código derivados del CxOne Standard Source Code and Construction⁴ (**Ver Anexo 1**). Este documento guía para desarrollar software de manera estandarizada fue provisto por la consultora informática Ace Consulting Services.

6.4.6.1 - ASP.NET

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de **.NET Framework** y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el **Common Language Runtime (CLR)**, entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, y todas las clases del Framework.

Ahora bien, veamos algunos conceptos antes mencionados:

- El **CLR** (Common Language Runtime) es el motor de ejecución de las aplicaciones .NET, lo que en Java sería la máquina virtual de Java, este motor se encarga de ejecutar todo el código .NET para ello ha de ser en dicho lenguaje. El CLR es el encargado de convertir este lenguaje intermedio en lenguaje máquina del procesador, esto normalmente se hace en tiempo real por un compilador JIT (Just-In-Time) que lleva incorporado el CLR.
- El conjunto de clases del **.NET Framework** es la piedra angular de cualquier desarrollador de .NET, es un rico conjunto de clases, interfaces, tipos que simplifican y optimizan el desarrollo de aplicaciones .NET además de proporcionar acceso a la funcionalidad del sistema. Como desarrolladores el dominio de este conjunto de clases es vital para un buen desarrollo en .NET.

⁴ <http://www.construx.com/Page.aspx?nid=79>

ASP.NET incluye:

- ✓ Marco de trabajo de página y controles
- ✓ Compilador de ASP.NET
- ✓ Infraestructura de seguridad
- ✓ Funciones de administración de estado
- ✓ Configuración de la aplicación
- ✓ Supervisión de estado y características de rendimiento
- ✓ Capacidad de depuración
- ✓ Marco de trabajo de servicios Web XML
- ✓ Entorno de host extensible y administración del ciclo de vida de las aplicaciones
- ✓ Entorno de diseñador extensible

Marco de trabajo de páginas y controles

El marco de trabajo de páginas y controles ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor Web para generar y representar de forma dinámica páginas Web ASP.NET. Las páginas Web ASP.NET se pueden solicitar a cualquier explorador o dispositivo del cliente y ASP.NET representa el marcado (como HTML) al explorador que realizó la solicitud. Como norma, puede utilizar la misma página para varios exploradores, porque ASP.NET representa el marcado adecuado para el explorador que realiza la solicitud. Sin embargo, puede diseñar una página Web ASP.NET para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 7, y aprovechar así todas las características de ese explorador. ASP.NET es compatible con los controles móviles de los dispositivos preparados para trabajar en Web como teléfonos celulares, PC portátiles y asistentes digitales personales (PDA).

Las páginas Web ASP.NET están completamente orientadas a objetos. En las páginas Web ASP.NET se puede trabajar con elementos HTML que usen propiedades, métodos y eventos. El marco de trabajo de páginas ASP.NET elimina los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página.

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Los controles se escriben una vez, se pueden utilizar en varias páginas y se integran en la página Web ASP.NET en la que se colocan durante la representación.

El marco de trabajo de páginas y controles ASP.NET también proporciona funciones para controlar la apariencia y el funcionamiento general de los sitios Web a través de temas y máscaras. Se pueden definir temas y máscaras y, a continuación, aplicarlos en las páginas o controles.

Además de los temas, es posible definir páginas principales que se crean para conseguir un diseño coherente en las páginas de la aplicación. Una página principal única define el diseño y el comportamiento estándar deseados para todas las páginas (o un grupo de páginas) de la aplicación. A continuación, se pueden crear páginas de contenido individuales con el contenido específico de la página que se desee mostrar. Cuando los usuarios solicitan las páginas de contenido, las combinan con la página principal con el fin de generar un resultado que combine el diseño de la página principal con el de la página de contenido.

Compilador de ASP.NET

Compila todo el código de ASP.NET, lo que permite el establecimiento inflexible de tipos, las optimizaciones de rendimiento y el enlace en tiempo de compilación, entre otras ventajas. Una vez que se ha compilado el código, el Common Language Runtime compila una vez más código de ASP.NET en código nativo, lo que permite un mayor rendimiento.

ASP.NET incluye un compilador que compilará todas las componentes de la aplicación, incluidas las páginas y los controles, en un ensamblado que el entorno de host de ASP.NET puede utilizar a continuación para atender las solicitudes del usuario.

Infraestructura de seguridad

Además de las características de seguridad de .NET, ASP.NET proporciona una infraestructura de seguridad avanzada para autenticar y autorizar el acceso de los usuarios y realizar otras tareas relacionadas con la seguridad. Puede autenticar usuarios con la autenticación de Windows suministrada por IIS (Internet Information

Service) o puede administrar la autenticación con su propia base de datos de usuario utilizando la autenticación mediante formularios ASP.NET y la suscripción ASP.NET. Además, puede administrar la autorización a las capacidades e información de su aplicación Web mediante los grupos de Windows o su propia base de datos de funciones personalizada utilizando las funciones de ASP.NET. Resulta fácil eliminar, agregar o reemplazar estos esquemas dependiendo de las necesidades de la aplicación.

Funciones de administración de estado

ASP.NET proporciona funcionalidad de administración de estado intrínseca que permite almacenar información entre las solicitudes de página, como la información de clientes o el contenido del carro de compra. Puede guardar y administrar información específica de la aplicación, específica de la sesión, específica de la página, específica del usuario y definida por el desarrollador. Esta información puede ser independiente de cualquier control de la página.

ASP.NET ofrece funciones de estado distribuidas, lo que le permite administrar información de estado en múltiples instancias de la misma aplicación en un equipo o en varios.

Configuración de ASP.NET

Las aplicaciones ASP.NET utilizan un sistema de configuración que le permite definir valores de configuración para su servidor Web, para un sitio Web o para aplicaciones individuales. Puede crear valores de configuración cuando se implementan las aplicaciones ASP.NET y puede agregar o revisar los valores de configuración en cualquier momento con un impacto mínimo en aplicaciones y servidores Web de operaciones. Los valores de configuración de ASP.NET se almacenan en archivos basados en la tecnología XML. Dado que estos archivos XML son archivos de texto ASCII, es fácil realizar cambios de configuración a sus aplicaciones Web. Puede extender el esquema de configuración para satisfacer sus requisitos.

Supervisión de estado y características de rendimiento

ASP.NET incluye características que le permiten supervisar el estado y el rendimiento de su aplicación ASP.NET. La supervisión del estado de ASP.NET permite proporcionar información sobre eventos clave que proporcionan información sobre el estado de una aplicación y sobre las condiciones de error. Estos eventos muestran una

combinación de diagnósticos y características de supervisión, a la vez que proporcionan un elevado grado de flexibilidad en lo que respecta a lo que se registra y cómo.

Capacidad de depuración

ASP.NET aprovecha la infraestructura de depuración en tiempo de ejecución para permitir la depuración entre lenguajes y equipos. Se pueden depurar tanto objetos administrados como no administrados, así como todos los lenguajes compatibles con el Common Language Runtime y los lenguajes de secuencias de comandos.

Además, el marco de trabajo de páginas ASP.NET proporciona un modo de seguimiento que permite insertar mensajes de instrumentalización en las páginas Web ASP.NET.

Marco de trabajo de servicios Web XML

ASP.NET es compatible con los servicios Web XML. Un servicio Web XML es un componente que incluye funcionalidad de empresa que permite a las aplicaciones intercambiar información entre firewalls utilizando estándares como los servicios de mensajería HTTP y XML. Los servicios Web XML no están relacionados con ninguna tecnología de componentes ni con ninguna convención de llamada a objetos en concreto. Como resultado, pueden obtener acceso a los servicios Web XML los programas escritos en cualquier lenguaje, que usen cualquier modelo de componentes y se ejecuten en cualquier sistema operativo.

Entorno de host extensible y administración del ciclo de vida de las aplicaciones

ASP.NET incluye un entorno de host extensible que controla el ciclo de vida de una aplicación desde el momento en que un usuario cualquiera tiene acceso a un recurso (como una página) en la aplicación hasta el momento en que se cierra la aplicación. Aunque ASP.NET se basa en un servidor Web (IIS) como un host de la aplicación, ASP.NET proporciona gran parte de la propia funcionalidad de host. La arquitectura de ASP.NET permite responder a los eventos de aplicación y crear controladores y módulos HTTP personalizados.

Entorno de diseñador extensible

ASP.NET incluye la compatibilidad mejorada para crear diseñadores de controles de servidor Web para utilizarlos con una herramienta de diseño visual como Visual Studio. Los diseñadores permiten crear una interfaz de usuario en tiempo de diseño para un control; de este modo, los desarrolladores pueden configurar las propiedades y el contenido del control en una herramienta de diseño visual.

6.4.6.2 - Web Services

Utilizaremos Web Services en nuestra solución para poder comunicarnos con nuestros clientes y así poder determinar si están online, es decir si están recibiendo las transacciones realizadas desde nuestra aplicación y eventualmente para transaccionar con aquellos clientes que deseen tener un Back Office propio. Esta característica será opcional para los clientes ya que la idea central es que la aplicación sea totalmente configurable por el usuario sin necesidad de intervención alguna de parte nuestra, pero bien, si el cliente desea minimizar el margen de problemas (cortes de luz o del servicio de internet) puede optar por instalar un web service en su back office, como así también mantener su propio repositorio de datos compartiendo únicamente la información necesaria para la operatoria del sitio.

Un **Servicio Web** es básicamente una función o procedimiento que puede ser accedida vía web por cualquier programa o aplicación sin importar en que plataforma reside el servicio o en que lenguaje ha sido desarrollado, el término "web" implica que el acceso se hace mediante una conexión a Internet habitualmente vía http aunque otros protocolos de transporte pueden ser utilizados. De esta forma, una aplicación Visual Basic puede, por ejemplo, usar una rutina de conversión de metros a yardas que fue realizada en Python y reside en un servidor de la India.

Esta interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Para que quede más claro veremos mediante un ejemplo como funciona un Servicio Web.

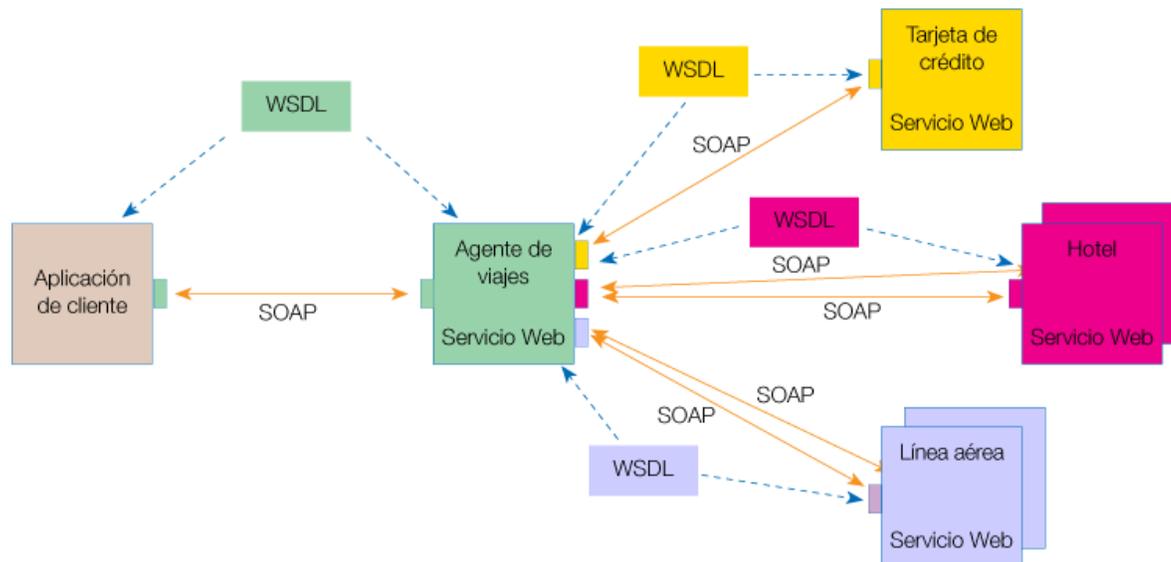


Figura 6. Ejemplo práctico de un servicio Web

Fuente: <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb>

Según el ejemplo del gráfico, un usuario (que juega el papel de cliente dentro de los Servicios Web), a través de una aplicación, solicita información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes que ofrece sus servicios a través de Internet. La agencia de viajes ofrecerá a su cliente (usuario) la información requerida. Para proporcionar al cliente la información que necesita, esta agencia de viajes solicita a su vez información a otros recursos (otros Servicios Web) en relación con el hotel y la compañía aérea. La agencia de viajes obtendrá información de estos recursos, lo que la convierte a su vez en cliente de esos otros Servicios Web que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes que servirá de intermediario entre el usuario y el servicio Web que gestionará el pago.

Trasladando este ejemplo a nuestra solución podemos ver cómo podríamos mediante nuestra aplicación llamar Servicios Web ubicados en nuestros clientes (negocios) para saber simplemente si están online o yendo un poco más allá para transaccionar con sus propios entornos (Back Office)

En todo este proceso intervienen una serie de tecnologías que hacen posible esta circulación de información. Por un lado, estaría **SOAP** (Protocolo Simple de Acceso a Objetos). Se trata de un protocolo basado en XML, que permite la interacción entre

varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP. SOAP especifica el formato de los mensajes. El mensaje SOAP está compuesto por un **envelope** (sobre), cuya estructura está formada por los siguientes elementos: **header** (cabecera) y **body** (cuerpo).

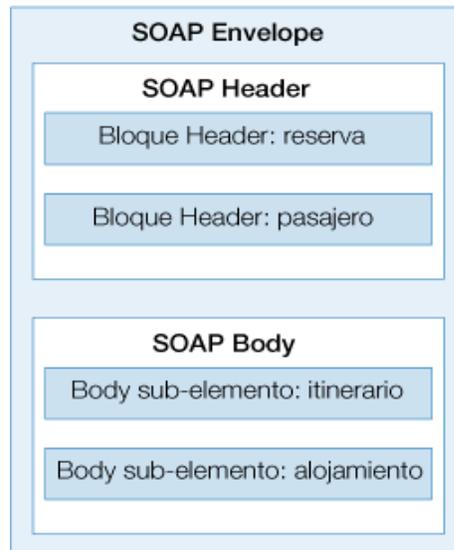


Figura 7. Servicios Web. Estructura de los Mensajes

Fuente: <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb>

Para optimizar el rendimiento de las aplicaciones basadas en Servicios Web, se han desarrollado tecnologías complementarias a SOAP, que agilizan el envío de los mensajes (MTOM) y los recursos que se transmiten en esos mensajes (SOAP-RRSHB).

Por otro lado, **WSDL** (Lenguaje de Descripción de Servicios Web), permite que un servicio y un cliente establezcan un acuerdo en lo que se refiere a los detalles de transporte de mensajes y su contenido, a través de un documento procesable por dispositivos. WSDL representa una especie de contrato entre el proveedor y el que solicita. WSDL especifica la sintaxis y los mecanismos de intercambio de mensajes.

6.4.7 Tecnologías utilizadas desde el lado del Cliente

6.4.7.1 JavaScript

Necesitamos un lenguaje que sea interpretado por los navegadores en su totalidad, ya que, estos serán el medio que utilizarán nuestros clientes para ejecutar la aplicación. Casi por decantación este lenguaje es **JavaScript**, todos los navegadores

interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Document Object Model). Javascript se puede incluir en cualquier documento HTML (o XHTML), o todo aquel que termine traducándose en HTML (o XHTML) en el navegador del cliente; ya sea PHP, ASP, SVG...

JavaScript permite que no se realicen pedidos innecesarios al servidor, sino que se puedan resolver algunas de las reglas de negocio directamente en el navegador haciendo más eficiente el uso de los recursos. Por ejemplo cuando se utiliza un formulario en el que se debe introducir un número en alguno de los campos y el usuario del sistema introduce erróneamente una letra, sin la utilización de JavaScript para controlar esto, se enviaría el formulario al servidor el cual identificaría que el tipo de dato introducido es incorrecto. Este último, entonces, deberá enviar nuevamente el formulario al cliente para que lo complete correctamente (informándole del error). Como se puede ver en el ejemplo, se ha producido una transferencia innecesaria entre el cliente y el servidor. Lo que JavaScript permite en un caso como este es verificar el tipo de datos introducido en el formulario antes de enviarlo al servidor y de producirse el error, este es detectado directamente en el navegador y se puede informar al usuario sin necesidad de enviar el formulario al servidor, hasta que este tenga los tipos de datos correctos para ser procesados.

Además, muchas de las peticiones al servidor utilizan una variación de JavaScript conocida como AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML Asíncronos), lo cual es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

6.4.7.2 AJAX

AJAX suele ser definido como un modelo web que permite crear aplicaciones que trabajan en forma asíncrona con el servidor, volcando la mayor parte del procesamiento del lado del cliente. Sobre esa idea inicial se sustenta, justamente, su denominación, que es un acrónimo de **Asynchronous Javascript And Xml**.

En AJAX las ventajas son muy evidentes, en cuanto a interactividad incide sustancialmente en la presentación de aplicaciones web, ya que estas no sólo ganan en rapidez sino que también ahorran un importante tiempo en la carga de datos, algo que no suele ocurrir aun en la más óptima de las aplicaciones clásicas.

En el vertiginoso crecimiento de AJAX influyó no sólo el adelanto de los principales navegadores web del mercado, que empezaron a soportar este tipo de programación en el último tiempo, sino también en las notorias mejoras de velocidad y estabilidad en las comunicaciones. Por eso, muchos definen AJAX como una técnica especial de desarrollo que, si bien esta presente desde hace algunos años, recién ahora ha sido considerada y perfeccionada.

Y es que en verdad la explosión de AJAX se dio en la medida que creció el soporte hacia el objeto XMLHttpRequest de JavaScript, el cual actúa como elemento de cohesión para un grupo de tecnologías que mejoran la interfaz de una aplicación web, trabajando siempre en forma conjunta e interactuando entre sí.

AJAX está formado por diversas tecnologías que interactúan. Pasemos a evaluar cuáles son esos componentes que permites definir en su totalidad una aplicación de este tipo:

JavaScript: Este famoso lenguaje de script es utilizado en AJAX para comunicarnos en tiempo y forma con el servidor, manipular el DOM⁵ (Document Object Model) e interactuar con todas las tecnologías disponibles, siendo el principal y más poderoso factor de cohesión. Ya desde el mismísimo objeto XMLHttpRequest (XHR), también denominado interfaz XMLHttpRequest para Internet Explorer, vemos como JavaScript se usa para manejar en forma nativa el propio corazón de AJAX, de esa manera se busca estar inmerso en la mayoría de los navegadores. Este objeto se encarga de de la comunicación y la recuperación asincrónica de datos con el servidor.

- **XHTML/CSS:** Toda capa de presentación estándar usa XHTML y CSS, aquí es donde estarán los controles que visualizará el usuario. Esto se complementa con un subconjunto dinámico inmerso en dicha capa, pivotado sobre DOM. Ese esquema basado en DOM gestiona los cambios dinámicos en el contenido de la presentación, y asiste en la gestión y en el control de eventos desencadenados por los usuarios.
- **XML:** Si bien la denominación AJAX nació desde su propio nombre con el concepto de XML para manipular e intercambiar datos entre el cliente y el

⁵ DOM o Document Object Model (en inglés, Modelo de Objetos de Documento). Es una Interfaz independiente de la plataforma y del lenguaje que permite a programas y scripts acceder y actualizar dinámicamente los contenidos, la estructura y el estilo de los documentos XHTML y XML.

servidor, esta pieza fundamental hoy parece más bien una alternativa válida frente a otras tecnologías cada vez más usadas, como es el caso de **JSON** (JavaScript Object Notation).

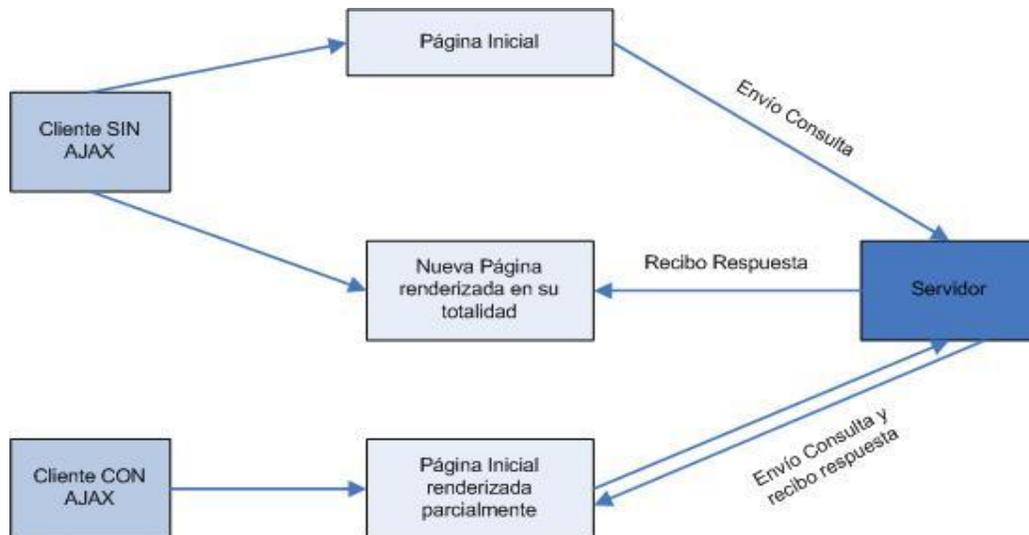


Figura 8. Comparación del Modelo Clásico con AJAX. Fuente: Revista Users .Code N° 31. Editorial MPEdiciones

6.4.8 - Base de Datos

Parte fundamental de nuestro proyecto es el almacenamiento, disponibilidad y seguridad de los datos, es por esto que el motor de base de datos seleccionado debe ser lo más completo y robusto posible. SQL Server 2005 cumple con estas características y si bien presenta una desventaja en cuanto al costo de sus licencias recordemos que el objetivo final de este proyecto no está orientado a sólo un cliente (con lo que tendríamos un presupuesto muy acotado) sino a desarrollar una solución que puedan adoptar con facilidad y de la misma manera muchos clientes, tantos como el mercado lo permita, por lo tanto y teniendo en cuenta que el volumen de transacciones y cantidad de información almacenada dependerá directamente de la cantidad de clientes que tengamos creemos que este motor de base de datos tendrá las mejores prestaciones.

Otro factor clave para tomar esta decisión es que el proyecto va a ser desarrollado con tecnologías Microsoft y SQL Server 2005 se adaptaría a la perfección por ser un producto de la misma empresa.

Por otro lado para los clientes que decidan tener su Back Office propio puede implementarse sin problema la versión SQL Express Edition, que no tiene costo alguno.

6.4.8.1 – SQL Server 2005

SQL Server es una solución de datos global, integrada y de extremo a extremo que habilita a los usuarios en toda una organización mediante una plataforma más segura, confiable y productiva para datos empresariales y aplicaciones de BI (Business Intelligence). SQL Server 2005 provee herramientas sólidas y conocidas a los profesionales de IT, así como también a trabajadores de la información, reduciendo la complejidad de la creación, despliegue, administración y uso de aplicaciones analíticas y de datos empresariales en plataformas que van desde los dispositivos móviles hasta los sistemas de datos empresariales. A través de un conjunto global de características, la interoperabilidad con sistemas existentes y la automatización de tareas rutinarias, SQL Server 2005 ofrece una solución completa de datos para empresas de todos los tamaños.

La plataforma de datos SQL Server incluye las siguientes herramientas:

- **Base de datos relacional.** Un motor de base de datos relacional más segura, confiable, escalable y altamente disponible con mejor rendimiento y compatible para datos estructurados y sin estructura (XML).
- **Servicios de réplica.** Réplica de datos para aplicaciones de procesamiento de datos distribuidas o móviles, alta disponibilidad de los sistemas, concurrencia escalable con almacenes de datos secundarios para soluciones de información empresarial e integración con sistemas heterogéneos, incluidas las bases de datos Oracle existentes.
- **Notification Services.** Capacidades avanzadas de notificación para el desarrollo y el despliegue de aplicaciones escalables que pueden entregar actualizaciones de información personalizadas y oportunas a una diversidad de dispositivos conectados y móviles.
- **Integration Services.** Capacidades de extracción, transformación y carga (ELT) de datos para almacenamiento e integración de datos en toda la empresa.
- **Analysis Services.** Capacidades de procesamiento analítico en línea (OLAP) para el análisis rápido y sofisticado de conjuntos de datos grandes y complejos, utilizando almacenamiento multidimensional.

- **Reporting Services.** Una solución global para crear, administrar y proporcionar tanto informes tradicionales orientados al papel como informes interactivos basados en la Web.
- **Herramientas de administración.** SQL Server incluye herramientas integradas de administración para administración y optimización avanzadas de bases de datos, así como también integración directa con herramientas tales como Microsoft Operations Manager (MOM) y Microsoft Systems Management Server (SMS). Los protocolos de acceso de datos estándar reducen drásticamente el tiempo que demanda integrar los datos en SQL Server con los sistemas existentes. Asimismo, el soporte del servicio Web nativo está incorporado en SQL Server para garantizar la interoperabilidad con otras aplicaciones y plataformas.
- **Herramientas de desarrollo.** SQL Server ofrece herramientas integradas de desarrollo para el motor de base de datos, extracción, transformación y carga de datos, minería de datos, OLAP e informes que están directamente integrados con Microsoft Visual Studio para ofrecer capacidades de desarrollo de aplicación de extremo a extremo. Cada subsistema principal en SQL Server se entrega con su propio modelo de objeto y conjunto de interfaces del programa de aplicación (API) para ampliar el sistema de datos en cualquier dirección que sea específica de su negocio.

7 – PROPUESTA

Se detalla en este punto como funcionará la solución propuesta. Para entender mejor el detalle de funcionalidad vamos a seguir el diagrama de la **Figura 9**, como se puede apreciar tenemos en primera medida grupos de usuarios divididos por rubro, estos son los usuarios finales del sistema, no son nuestros clientes sino que son los clientes de nuestros clientes. Los dividimos por rubro porque cada rubro tendrá su propia Sitio Web, como aclaramos anteriormente a los fines de este trabajo final únicamente nos enfocaremos en el rubro **deportes**, pero quedarían sentadas las bases y la estructura para implementar nuevos rubros.

Este sitio web destinado a los usuarios finales por rubro, tendrá un motor de búsquedas por diferentes parámetros, como ser para el caso del rubro deportes, tipo de deporte, ubicación y disponibilidad de horarios. Estas búsquedas son posibles ya

que los datos están centralizados en una única base. Una vez que el usuario mediante las búsquedas determine qué espacio desea reservar comienza el proceso de reservación. Para realizar la reservación el usuario debe loguearse en el sistema, una vez logueado envía la solicitud y la Aplicación verifica las reglas de negocio configuradas para el cliente requerido, además, opcionalmente vía Web Service verifica que el cliente este online para asegurarse de alguna manera que la solicitud se esté recibiendo. El usuario recibirá la respuesta del cliente dependiendo las reglas de negocio y por los canales (email, sms) previamente configurados para este usuario.

Hasta aquí vimos básicamente el comportamiento de la Aplicación desde el punto de vista de los usuarios finales, con un Sitio Web según el rubro, pero por otro lado debemos brindar una solución a nuestros propios clientes, es decir, los negocios que ponen sus espacios en alquiler. Esta solución es también una Aplicación Web, pero a diferencia de la anterior esta es centralizada, es decir no se divide por rubros, ni por ningún otro parámetro, es una sola para todos nuestros clientes. El objetivo de este módulo es que los negocios administren las reservas de sus espacios, para ello inicialmente deberán configurar que tipo de espacio ofrecen, cantidad, horarios y días de atención, precios. Y por supuesto actualizar en tiempo real las reservas que se realicen por otros canales (telefónico, personal, mensajería), toda esta información se centralizará en nuestra base de datos SQL Server 2005 y será fundamental para mantener consistente y disponible los datos para las consultas de los sitios Web para Usuarios. Los clientes a su vez, pueden optar por contar con una base de datos local propia como respaldo por si pierden conectividad (Internet) y así poder mantener los datos.

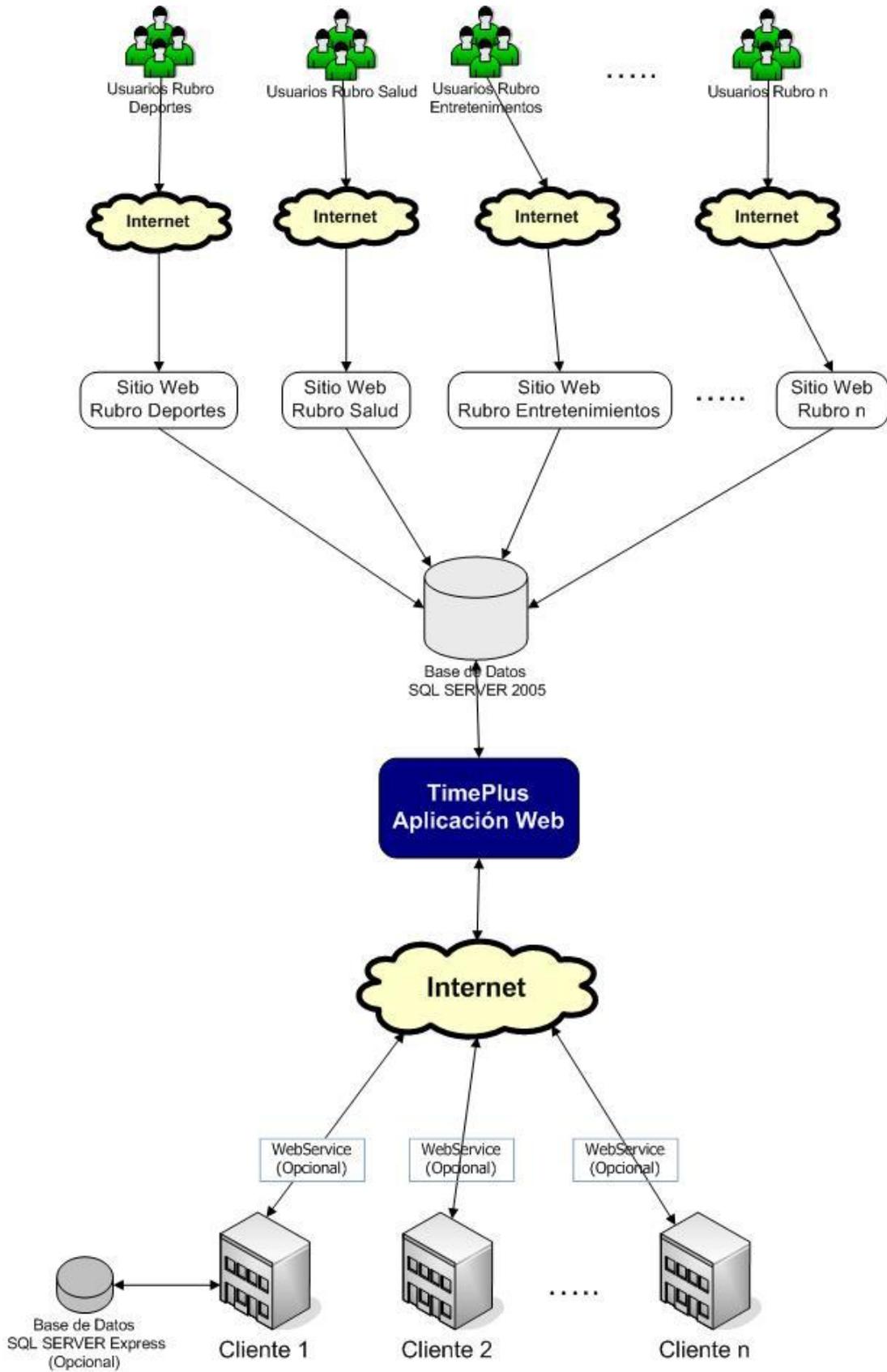


Figura 9. Esquema General del Proyecto

8 – ESTIMACIÓN Y PRESUPUESTO

Se realizará la planificación y estimación de tiempo para el diseño, desarrollo e implementación (puesta en línea) de la solución. Conjuntamente se calcularon los costos asociados, hay que tener en cuenta que esta es una estimación aproximada y realizando algunos supuestos, ya que este trabajo final de graduación es unipersonal todas las actividades las realiza la misma persona. Los supuestos son que de acuerdo a la actividad se contrata a un especialista específico que se encargue de la tarea sumado a un Jefe de proyecto que hará el seguimiento integral del desarrollo e implementación. Se considera que el Jefe de Proyecto tiene un sueldo mensual fijo de \$4000 que se prorratearán a lo largo del desarrollo, en cambio los demás profesionales son contratados por hora.

Tomando como referencia el cuadro de honorarios del Consejo de Ciencias Informáticas de la Provincia de Córdoba, se determinará en base a la estimación el precio aproximado de la solución. **Ver Anexo 1.**

Según el cuadro de la Figura 10 el trabajo tiene un costo aproximado de \$42080 en recursos humanos, a lo que hay que sumarle costos de hosting mensuales que no son significantes.

Este sería el precio aproximado de venta con el trabajo finalizado si alguien quisiera explotarlo. La solución está pensada para explotarse cobrando mensualidades que se cobrarían a los negocios o empresas captadas, de acuerdo a los espacios de alquiler que posean.

Actividades	Estimación	Profesional	Precio Hora	Precio Total
Fundamentación del Trabajo	4 Días	Analista Funcional	50	1600
Determinación de Objetivos, Alcance y Límite	6 Días	Analista Funcional	50	1600
Planificación	5 Días	Analista Funcional	50	2000
Diseño de Diagramas de Análisis				
Modelo de Contexto	2 Días	Analista Funcional	50	800
Modelo de Eventos	8 Días	Analista Funcional	50	3200
Modelo de Información	5 Días	Analista Funcional	50	2000
Diseño de Interfaz y Prototipo de Navegabilidad	6 Días	Diseñador Gráfico	25	1200
Diseño de Base de Datos	5 Días	Administrador de Base de datos	50	2000
Programación				
Registrar Nueva Entidad	3 Días	Programador de Páginas Web	30	720
Registrar Operador Administrador de Entidad	2 Días	Programador de Páginas Web	30	480
Inicio de Sesión	2 Días	Programador de Páginas Web	30	480
Registro inicial de Espacios	5 Días	Programador de Páginas Web	30	1200
Gestión de Espacios por Entidad	4 Días	Programador de Páginas Web	30	960
Gestión de Horarios por Espacio	3 Días	Programador de Páginas Web	30	720
Gestión de Días no Laborables	1,5 Días	Programador de Páginas Web	30	360
Gestión de Operadores	3 Días	Programador de Páginas Web	30	720
Gestión de Prestadores	3 Días	Programador de Páginas Web	30	720
Visualización de Agenda	7 Días	Programador de Páginas Web	30	1680
Bandeja de Entrada de Autorización de Turnos	5 Días	Programador de Páginas Web	30	1200
Carga de Turnos por Otros canales	4 Días	Programador de Páginas Web	30	960
Modificar/Eliminar Turnos registrados	3 Días	Programador de Páginas Web	30	720
Gestión de Clientes	3 Días	Programador de Páginas Web	30	720
Registrar Nuevo usuario	2,5 Días	Programador de Páginas Web	30	600
Iniciar sesión de usuario	2 Días	Programador de Páginas Web	30	480
Consultar Disponibilidad de Turnos	5 Días	Programador de Páginas Web	30	1200
Solicitar Turno	4 Días	Programador de Páginas Web	30	960
Administrar Turnos Solicitados	3 Días	Programador de Páginas Web	30	720
Configurar Datos propios	2 Días	Programador de Páginas Web	30	480
Testing	6 Días	Analista Funcional	50	1600
Implementación	2 Días	Jefe de proyecto		
	116 Días			\$32080
Salario de Jefe de Proyecto	5 meses al 50%		4000	10000
TOTAL				\$42080

Figura 10. Estimación y Costo de la solución

9 – ANÁLISIS

9.1 – Plan General

Según lo establecido en el marco teórico el modelo a utilizar para realizar el análisis funcional del sistema es el propuesto por David A. Ruble en su libro "Análisis y Diseño Práctico de Sistemas Cliente/Servidor con GUI".

Como primera premisa Ruble establece que se debe definir el "Plan General del Proyecto", para ello lo primero es definir una Meta, que es donde apunta el proyecto, su razón de ser y que se quiere lograr en el largo plazo, la meta debe ser clara, no ambigua, concisa y mensurable, todo el equipo del proyecto debe conocerla a la perfección y saber cuándo se ha alcanzado. Podemos establecer como meta de nuestro proyecto *"Desarrollar un sistema lo suficientemente general y configurable con el cual podamos captar los clientes de determinado segmento de mercado (según el rubro) para que éstos le brinden a sus clientes un nuevo canal de comunicación eficiente para realizar procesos de reservación"*. Este enunciado tiene una connotación distinta del establecido como objetivo en el punto 6, esto se debe a que esta meta esta enunciada desde el punto de vista del negocio como empresa lucrativa y este es el fin último, *"Desarrollar un sistema tan útil y eficiente que se venda sólo"*

Luego de establecida la meta, Ruble establece que deben definirse los objetivos. Los objetivos se determinan descubriendo todos los problemas que hay en la forma actual de hacer negocios y proporcionando ideas sobre oportunidades no aprovechadas, así los objetivos son las actividades específicas que se deben desarrollar para alcanzar la meta. Si bien podemos identificar algunos problemas en la operatoria diaria de nuestro cliente (Tancat Deportes) como ser, la falta de disponibilidad y respaldo de los datos e ineficiente asignación de reservas entre sucursales; este proyecto se orienta más a aprovechar una oportunidad, como dice Ruble, una oportunidad se presenta cuando se puede aprovechar una nueva tecnología, productos o servicios (aquí entraría nuestro proyecto, como un servicio novedoso, que usa una tecnología relativamente nueva como es Internet) que no existían antes o que no habían sido considerados.

Un objetivo es una frase que cuando se lleva a cabo elimina el problema o aprovecha una oportunidad. Podemos identificar algunos objetivos para nuestro proyecto

- ✓ Diseñar un Sitio Web para los usuarios dinámico e interesante para poder captarlos, ya que estos son los motores del proyecto.

- ✓ Que el nuevo canal de comunicación proporcionado (Internet) entre los usuarios y los negocios tenga un crecimiento exponencial de su uso en el tiempo.
- ✓ Que los negocios puedan organizar el alquiler sus espacios de manera eficiente, maximizando las reservas posibles.
- ✓ Disminuir las reservas no efectivizadas con un sistema de alertas para el usuario por varios canales (email, sms), además de la identificación e historial de comportamiento de mismo.

Por último debemos definir para el plan del proyecto, los criterios de evaluación y opciones de solución. Los criterios de evaluación establece la manera en que se medirá la solución dada en relación con los objetivos, estos criterios serán establecidos una vez implementado el sistema para poder medir y comparar las variables cuantificables identificadas en los objetivos.

Las opciones de solución plantean medir el costo/beneficio de hacer algo contra la línea base de no hacer nada. Mantener la operatoria como está actualmente (sin ningún sistema informático) es nuestra alternativa, podremos definir si el proyecto fue exitoso cuando tengamos los valores de los criterios de evaluación disponibles para los negocios que se realice el prototipo.

Aprovechamos aquí para aclarar un punto, una vez desarrollado el sistema será implementado en nuestro cliente (Tancat Deportes) y en algún otro cliente medible, a partir de allí comenzaremos obtener métricas y la usabilidad real del sistema, para finalmente determinar la viabilidad del mismo en otros clientes.

9.2 – Modelo de Contexto

El contexto representa todo el modelo del proceso

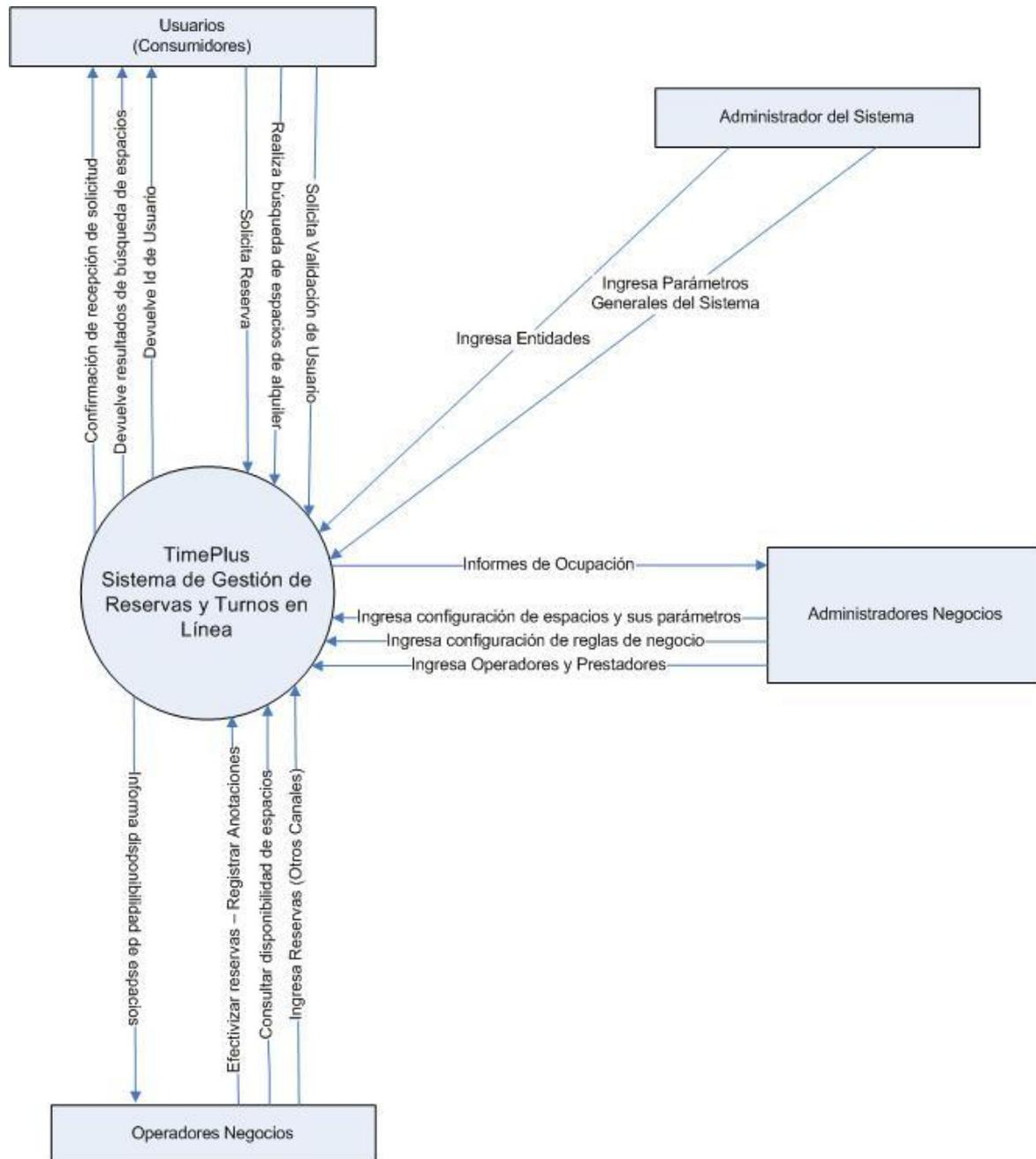


Figura 10. Modelo de Contexto. Nivel 0

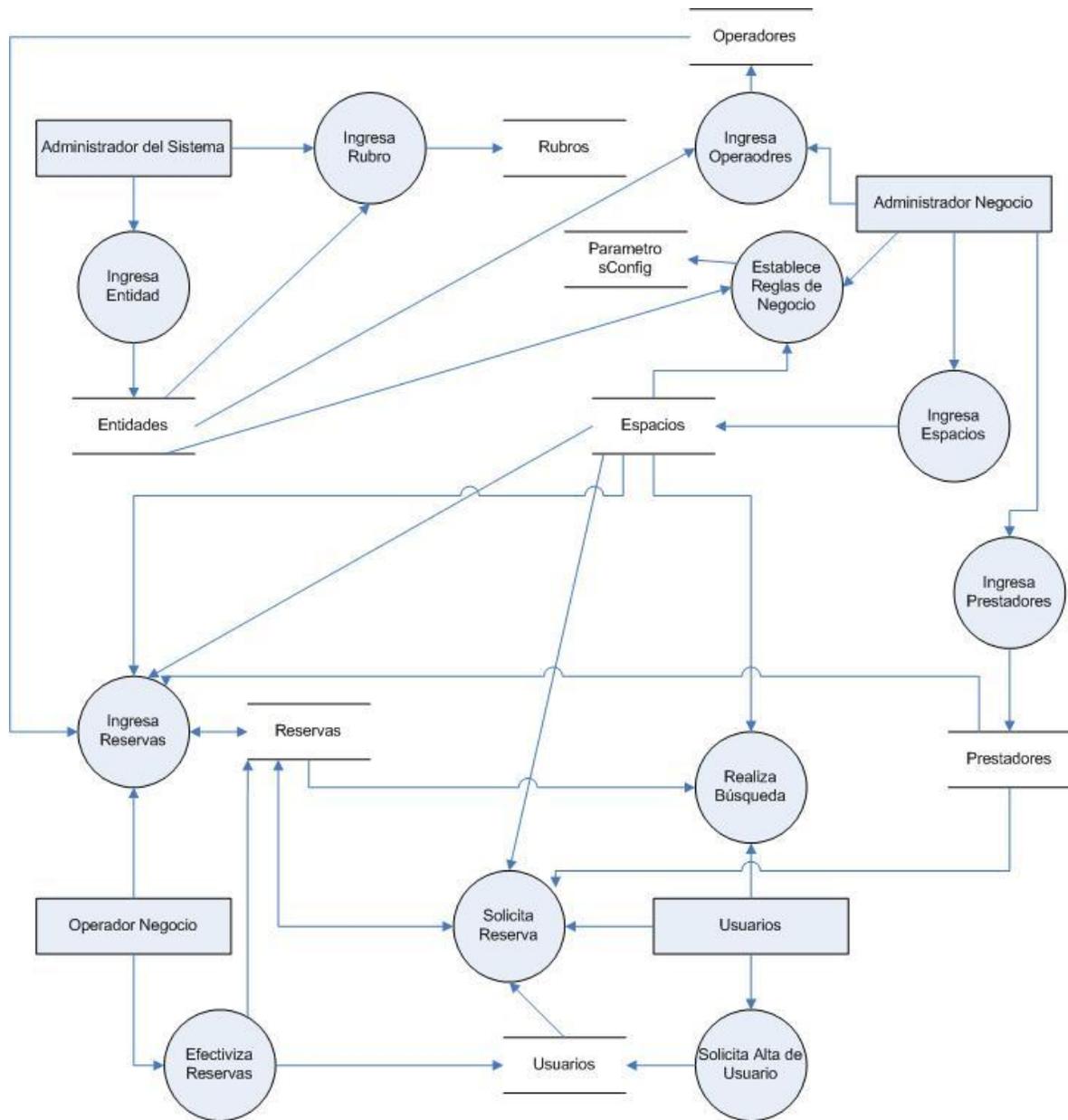


Figura 11. Modelo de Contexto. Nivel 1

9.3 – Modelo de Eventos

Define los requerimientos del sistema. Para ello dispone de dos herramientas: la lista de eventos y el diccionario de eventos:

- Lista de Eventos: cataloga cada evento que el sistema debería contemplar con una sintaxis sujeto-verbo-objeto
- Diccionario de Eventos: resulta más útil tanto para el analista como para el diseñador ya que detalla un evento en particular en Estímulo → Actividad → Respuesta.

9.3.1 – Lista de Eventos

Módulo Administrativo

1. Registrar Nueva Entidad
2. Registrar Operador Administrador de Entidad
3. Iniciar Sesión
4. Registro inicial de espacios
5. Gestión de Espacios por Entidad
6. Gestión de Horarios por Espacio
7. Gestión de Días no laborables
8. Gestión de Operadores
9. Gestión de Prestadores

Módulo Trabajo

10. Visualización de Agenda
11. Bandeja de entrada de autorización de turnos
12. Carga de turnos por otros canales
13. Modificar/Eliminar turnos registrados
14. Gestión de Clientes

Módulo Usuario Final

15. Registrar nuevo usuario
16. Iniciar sesión de usuario
17. Consultar disponibilidad de turnos
18. Solicitar turno
19. Administrar turnos solicitados
20. Configurar datos propios

9.3.2 – Diccionario de Eventos

Resumen de Roles

ROL	DESCRIPCION
Administrador de Sistema	Configura los parámetros y tablas de configuración de la aplicación
Operador Administrador de Entidad	Tiene permisos sobre todas las funcionalidades del sistema desde el punto de vista del negocio. Configura espacios, operadores y prestadores. y además administra y gestión turnos en agenda
Operador de Entidad	Tiene acceso sólo a las páginas pertinentes a Administración y Gestión de turnos de agenda
Usuario Final	No accede al sistema TimePlus directamente sino que a través de los sitios particulares de un negocio y algún sitio general de rubro. Es quién solicita y consume turnos en espacios de alquiler.

CU – Registrar Nueva Entidad

Nombre del Use Case: Registrar Nueva Entidad		Nro. de Orden: 01
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad (o Usuario nuevo)	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Registrar en la aplicación una nueva empresa, organización, entidad o persona		
Precondiciones: No Aplica		
Post- Condiciones: La datos de la entidad deben ser correctos y mantenidos en memoria		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario (por lo general quién tendrá el rol de Administrador de entidad) desea registrarse en el sistema e ingresa en la opción "Registrarse"		
El Sistema presenta un formulario de carga de datos para la nueva entidad.		
El usuario Administrador de Entidad ingresa los datos correspondientes a su entidad y presiona la opción "Siguiete"		
El sistema valida los datos ingresados y verifica que se hayan cargado correctamente todos los datos requeridos.	4.1.- El sistema comprueba que hay datos requeridos no ingresados o algún dato no ingresado correctamente. 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.	
El Sistema mantiene en memoria los datos cargados		
El CU continúa en "02 – Registrar Operador Administrador de Entidad"		
Fin de UC.		
Use Case donde se incluye:		
Use Case al que extiende: Registrar Operador Administrador de Entidad		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Registrar Operador Administrador de Entidad

Nombre del Use Case: Registrar Operador Administrador de Entidad		Nro. de Orden: 02
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	

Actor Principal: Administrador de Entidad (o Usuario nuevo)		Actor Secundario:	
Tipo de Use Case: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
Objetivo: Registrar los datos del Operador Administrador de Entidad			
Precondiciones: El usuario debe previamente haber cargado los datos de la entidad			
Post- Condiciones: El Sistema debe registrar los datos de la entidad y del Operador Administrador de Entidad			
Curso Normal		Alternativas	
El Caso de Uso se inicia cuando es llamado por el CU "01 – Registrar Nueva Entidad"			
El Sistema presenta un formulario de carga de datos para el usuario Operador Administrador de Entidad.			
El usuario carga los datos y presiona la opción "Registrar"			
El Sistema verifica que la carga de datos es correcta y que los datos requeridos fueron ingresados		4.1.- El sistema comprueba que hay datos requeridos no ingresados o algún dato no ingresado correctamente. 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.	
El sistema registra los datos de Nueva Entidad que fueron mantenidos en memoria en el CU anterior y registra los datos del usuario Operador Administrador de Entidad.			
Fin de UC.			
Use Case donde se incluye: 01 – Registrar Nueva Entidad (Paso 1)			
Use Case al que extiende:			
Use Case de Generalización:			
Autor: Bortagaray, Martín		Fecha Creación: 11/02/2009	
Autor Última Modificación:		Fecha Última Modificación:	

CU – Iniciar Sesión

Nombre del Use Case: Iniciar Sesión		Nro. de Orden: 03	
Prioridad: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable			
Complejidad: <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Medio <input type="checkbox"/> Simple <input type="checkbox"/> Muy Simple			
Actor Principal: Administrador del Sistema, Administrador de Entidad, Operador de Entidad		Actor Secundario:	
Tipo de Use Case: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
Objetivo: Validar el ingreso del usuario a la aplicación, cargando los permisos correspondientes a cada rol			
Precondiciones: El usuario debe estar registrado en el sistema			
Post- Condiciones: El sistema debe validar el ingreso y cargar los permisos correspondientes			
Curso Normal		Alternativas	
El Caso de Uso se inicia cuando un usuario desea			

identificarse unívocamente en el sistema e ingresa en "Iniciar Sesión"	
El Sistema presenta una pantalla donde solicita el ingreso de la Entidad a la que corresponde el usuario, su nombre de usuario y su contraseña.	
El usuario ingresa los datos y presiona "Aceptar"	
El Sistema verifica que la carga de datos es correcta.	4.1.- El sistema comprueba que hay datos ingresados incorrectamente 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.
El Sistema registra el ingreso del usuario operador y carga los permisos correspondientes.	
Fin de UC.	
Use Case donde se incluye: 01 – Registrar Nueva Entidad (Paso 1)	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Registro Inicial de Espacios

Nombre del Use Case: Registro Inicial de Espacios		Nro. de Orden: 04
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Administrador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Registrar por lo menos un espacio para que la entidad pueda comenzar a operar		
Precondiciones: El usuario debe estar logueado en el sistema con el rol "Operador Administrador de Entidad"		
Post- Condiciones: Se deben registrar el o los espacios para la entidad		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario Operador Administrador de Entidad ingresa al sistema y su entidad no tiene cargado ningún espacio de alquiler		
El Sistema presenta un formulario en una pestaña de carga de nuevo espacio donde solicita datos generales como tipo de espacio, descripción, frecuencia de agenda, si tiene asociado un prestador y si está activo.		
El Sistema presenta en otra pestaña un formulario de carga de días / horarios para este nuevo espacio.		
El usuario carga los datos correspondientes y presiona la opción "Crear Espacio"		

El Sistema verifica que la carga de datos es correcta y que todos los datos requeridos fueron cargados.	4.1.- El sistema comprueba que hay datos ingresados incorrectamente o datos requeridos que no fueron cargados 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 4.
El Sistema registra los datos correspondientes al nuevo espacio y horarios.	
Fin de UC.	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Gestión de Espacios por Entidad

Nombre del Use Case: Gestión de Espacios por Entidad		Nro. de Orden: 05
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Mantener espacios de alquiler		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad, por lo tanto la Entidad debe estar debidamente registrada en el sistema.		
Post- Condiciones: La Entidad para poder operar debe tener al menos una espacio configurado por entidad		
Curso Normal	Alternativas	
1.- El Caso de Uso se inicia cuando el Operador Administrador de Entidad ingresa en la opción "Gestión de Espacios"		
El Sistema presenta un listado con los espacios cargados y configurados para la entidad y presenta además las opciones de "Nuevo", "Editar", "Eliminar".		
El usuario selecciona la opción "Nuevo"	3.1.- El usuario selecciona un espacio del listado y presiona la opción "Editar" 3.1.1.- El sistema presenta los datos del espacio en un formulario y permite al usuario modificar ciertos datos. 3.1.2.- El usuario modifica los datos necesarios y presiona la opción "Aceptar" 3.1.3.- El CU continúa en el punto 6 3.2.- El usuario selecciona un espacio del listado y presiona la opción "Eliminar" 3.2.1.- El Sistema pide una confirmación y el proceso es confirmado. 3.2.2.- El Sistema verifica que el espacio de alquiler no tenga reservas activas asociadas.	

	3.2.3.- El Sistema da la baja lógica al espacio seleccionado 3.2.4.- Fin de UC
El Sistema presenta un formulario de carga de datos para el nuevo espacio como ser (Tipo de Espacio, Descripción, Frecuencia, Prestador Asociado y si esta activo.)	
El usuario carga los datos y presiona "Aceptar"	
El sistema verifica que los datos estén correctamente ingresados y que hayan sido ingresados todos los datos requeridos.	6.1.- El sistema comprueba que hay datos requeridos no ingresados o algún dato no ingresado correctamente. 6.1.1.- El Sistema informa los errores. 6.1.2.- CU continúa en punto 5.
El sistema registra los datos del nuevo espacio.	
Fin de UC.	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Gestión de Horarios por Espacio

Nombre del Use Case: Gestión de Horarios por Espacio		Nro. de Orden: 06
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Asignar horarios de disponibilidad a espacios de una entidad		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad, por lo tanto la Entidad debe estar debidamente registrada en el sistema y debe contener al menos un espacio de alquiler cargado		
Post- Condiciones: Deben quedar correctamente configurados los horarios de agenda para un espacio		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el Operador Administrador de Entidad ingresa en la opción "Gestión de Horarios por Espacio"		
El sistema presenta un listado con todos los espacios cargados para la entidad.		
El usuario selecciona un espacio		
El Sistema muestra un listado con los días / horarios configurados para el espacio y para cada ítem se presentan las opciones de "Editar" y "Eliminar". Por		

otra parte el sistema muestra la opción "Nuevo"	
El usuario selecciona un ítem y presiona la opción "Editar".	<p>5.1.- El usuario selecciona un ítem y presiona la opción "Eliminar"</p> <p>5.1.1.- El Sistema pide confirmación de la acción</p> <p>5.1.2.- El Sistema realiza la baja del día / horario seleccionado.</p> <p>5.1.3.- Fin de CU</p> <p>5.2.- El usuario no selecciona algún ítem y presiona la opción "Nuevo"</p> <p>5.2.1.- El Sistema permite el ingreso de un nuevo registro día / horario para el espacio seleccionado.</p> <p>5.2.2.- El Usuario ingresa los datos correspondientes y presiona "Aceptar"</p> <p>5.2.3.- Fin de UC</p>
El Sistema permite la modificación del día y/o horario.	
El usuario edita los valores deseados y presiona "Aceptar"	
El sistema registra los cambios ingresados	
Fin de CU	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Gestión de Días no Laborables

Nombre del Use Case: Gestión de Días no Laborables		Nro. de Orden: 07
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Administrador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Mantener el listado de días en los cuáles un espacio de una entidad no trabajará.		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad.		
Post- Condiciones: El Sistema debe registrar los días no laborables por espacio		

Curso Normal	Alternativas
El Caso de Uso se inicia cuando el Operador Administrador de Entidad ingresa en la opción "Gestión de Días no laborables"	
El sistema presenta un listado con todos los espacios cargados para la entidad.	
El usuario selecciona un espacio.	
El Sistema muestra un listado con los días y horarios no laborables para el espacio seleccionado si es que hubiera. El sistema presenta para cada ítem las opciones "Editar" y "Eliminar", además de la opción general "Nuevo" día no laborable	
El usuario selecciona un ítem y presiona la opción "Editar".	<p>5.1.- El usuario selecciona un ítem y presiona la opción "Eliminar"</p> <p>5.1.1.- El Sistema pide confirmación de la acción</p> <p>5.1.2.- El Sistema realiza la baja del día / horario seleccionado.</p> <p>5.1.3.- Fin de CU</p> <p>5.2.- El usuario no selecciona algún ítem y presiona la opción "Nuevo"</p> <p>5.2.1.- El Sistema permite el ingreso de un nuevo registro día / horario no laborable para el espacio seleccionado.</p> <p>5.2.2.- El Usuario ingresa los datos correspondientes y presiona "Aceptar"</p> <p>5.2.3.- Fin de UC</p>
El Sistema permite la modificación del día y/o horario no laborable.	
El usuario edita los valores deseados y presiona "Aceptar"	
El sistema registra los cambios ingresados.	
Fin de CU	
Asociaciones de Extensión:	
Asociaciones de Inclusión:	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Gestión de Operadores

Nombre del Use Case: Gestión de Operadores		Nro. de Orden: 08
Prioridad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Mantener el registro de Operadores por entidad		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad.		
Post- Condiciones: El Sistema debe registrar las modificaciones o nuevos Operadores ingresados		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el Operador Administrador de Entidad ingresa en la opción "Gestión de Operadores"		
El sistema presenta un listado con todos los Operadores cargados para la Entidad, con las opciones de "Editar" y "Eliminar" para cada ítem. Además el sistema presenta la opción general de registrar un "Nuevo Operador"		
El usuario selecciona un ítem y presiona la opción "Editar".	3.1.- El usuario selecciona un ítem y presiona la opción "Eliminar" 3.1.1.- El Sistema pide confirmación de la acción 3.1.2.- El Sistema realiza la baja lógica del Operador seleccionado siempre y cuando este no sea el único Operador Administrador de Entidad 3.1.3.- Fin de CU 3.2.- El usuario no selecciona algún ítem y presiona la opción "Nuevo" 3.2.1.- El Sistema permite el ingreso de un nuevo registro de un nuevo Operador para la entidad 3.2.2.- El Usuario ingresa los datos correspondientes y presiona "Aceptar" 3.2.3.- El Sistema registra el nuevo Operador 3.2.4.- Fin de UC	
El Sistema permite la modificación de los datos del Operador		
El usuario edita los valores deseados y presiona "Aceptar"		
El sistema registra los cambios ingresados.		
Fin de CU		

Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Visualización de Agenda de Espacios

Nombre del Use Case: Visualización de Agenda de Espacios		Nro. de Orden: 10
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Administrador de Entidad, Operador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Pantalla maestra de visualización de agenda de turnos por espacio		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad u Operador de Entidad		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el usuario ingresa en el sistema y la entidad ya tiene al menos un espacio completamente configurado.	1.1.- El Caso de Uso se inicia cuando el usuario ingresa en la opción "Gestión de Turnos"	
El sistema presenta un listado con los espacios disponibles para la entidad activa.		
El sistema presenta un calendario.		
El usuario selecciona un espacio y un día determinado		
El sistema despliega una agenda de turnos para la entidad y el día seleccionado.		
El usuario selecciona un casillero de la agenda y este contiene un registro.		
El Sistema abre un formulario con los datos del detalle de la reserva seleccionada		
Fin de CU		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Bandeja de Entrada de Autorización de Turnos

Nombre del Use Case: Bandeja de Entrada de Autorización de Turnos		Nro. de Orden: 11
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad, Operador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Proveer una interfaz de aceptación o rechazo de turnos		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad u Operador de Entidad		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el usuario ingresa en la opción "Autorización de Turnos" o ingrese mediante las alertas en la página de visualización de agenda		
El sistema presenta un listado de turnos ingresados que requieren autorización según la configuración propia de espacio.		
El sistema presenta para cada ítem de turnos las opciones de "Aceptar" y "Rechazar"		
El usuario selecciona un turno y presiona la opción "Aceptar".	4.1.- El usuario selecciona un turno y presiona la opción "Rechazar" 4.1.1.- El Sistema abre un formulario para que se cargue el motivo de rechazo. 4.1.2.- El usuario carga el motivo de rechazo y presiona "Aceptar" 4.1.3.- El Sistema quita la reserva de la agenda. 4.1.4.- El Sistema informa al usuario del rechazo y su motivo mediante los canales configurados (email y/o SMS) 4.1.5.- Fin de UC.	
El sistema registra el turno como aceptado y lo incluye en la agenda correspondiente.		
El Sistema informa al usuario que su turno ha sido autorizado mediante canales configurados (email y/o sms)		
Fin de CU		
Use Case donde se incluye:		
Use Case al que extiende:		

Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Carga de turnos por otros canales

Nombre del Use Case: Carga de turnos por otros canales		Nro. de Orden: 12
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Administrador de Entidad, Operador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Ingresar reservas de turnos mediante otros canales distintos a TimePlus como se presencial, telefónico o chat.		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad u Operador de Entidad		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el usuario desde la pantalla de “Visualización de Agenda” ingresa en un turno vacío.		
El sistema abre un formulario de carga de nuevo turno.		
El usuario carga los datos		
El Sistema verifica que los datos fueron cargados correctamente y que todos los datos requeridos fueron ingresados	4.1.- El sistema comprueba que hay datos ingresados incorrectamente o datos requeridos que no fueron cargados 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.	
El Sistema registra el nuevo turno y lo incluye en la agenda del espacio correspondiente.		
Fin de CU		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Modificar/Eliminar turnos registrados

Nombre del Use Case: Modificar/Eliminar turnos registrados		Nro. de Orden: 13
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Medio <input type="checkbox"/> Simple <input type="checkbox"/> Muy Simple	
Actor Principal: Administrador de Entidad, Operador de Entidad	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Modificar / Eliminar un turno ya registrado por cualquier canal.		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad u Operador de Entidad		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el usuario desde la pantalla de "Visualización de Agenda" ingresa en un ítem que tenga cargado algún registro (turno)		
El sistema abre un formulario con los datos del turno correspondiente, con las opciones de "Editar" y "Eliminar"		
El usuario presiona la opción "Editar"	3.1.- El usuario presiona la opción "Eliminar" 3.1.1.- El Sistema pide la verificación de la baja del turno seleccionado. 3.1.2.- El Sistema realiza la baja del turno seleccionado y dispara las notificaciones por los canales configurados si es que hubiese. 3.1.3.- Fin de CU.	
El Sistema habilita los campos posibles de edición.		
El usuario modifica los datos que desea y presiona "Aceptar"		
El Sistema verifica que la carga de datos sea correcta.	6.1.- El sistema comprueba que hay datos ingresados incorrectamente 6.1.1.- El Sistema informa los errores. 6.1.2.- CU continúa en punto 5.	
El Sistema registra los cambios en el turno y vuelve a la pantalla de "Visualización de Agenda" mostrando los cambios realizados.		
El Sistema dispara las notificaciones por los canales configurados si es que hubiere.		
Fin de CU		
Use Case donde se incluye:		
Use Case al que extiende:		

Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Gestión de Clientes

Nombre del Use Case: Gestión de Clientes		Nro. de Orden: 14
Prioridad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
Complejidad: <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Medio <input type="checkbox"/> Simple <input type="checkbox"/> Muy Simple		
Actor Principal: Administrador de Entidad, Operador de Entidad		Actor Secundario:
Tipo de Use Case: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Objetivo: Mantener los clientes propios de la Entidad		
Precondiciones: El usuario debe estar logueado en el sistema con el rol Operador Administrador de Entidad u Operador de Entidad		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando el usuario ingresa en la opción "Gestión de Clientes"		
El Sistema muestra una pantalla con un listado de todos los clientes de TimePlus que muestra las estadísticas de cumplimiento de cada cliente y una serie de filtros de búsqueda.		
El Sistema presenta la opción para los clientes de la Entidad de darles de baja y el usuario selecciona esta opción	<p>3.1.- El Sistema presenta la opción de dar de Alta un cliente para la entidad y el usuario selecciona esta opción.</p> <p>3.1.1.- El Sistema pide el ingreso de tipo y número de documento del cliente.</p> <p>3.1.2.- El usuario ingresa Tipo y Número de documento del cliente.</p> <p>3.1.3.- El Sistema verifica si este usuario ya está registrado en TimePlus y ya está registrado.</p> <p>3.1.3.1.- El Sistema verifica si este usuario ya está registrado en TimePlus y no está registrado.</p> <p>3.1.3.2.- El sistema abre un formulario de Nuevo cliente y pide el ingreso de los datos correspondientes</p> <p>3.1.3.3.- El usuario ingresa los datos.</p> <p>3.1.3.4.- El sistema registra los datos</p>	

	del nuevo cliente. 3.1.4.- El Sistema registra al usuario como cliente de la Entidad activa
El Sistema pide una confirmación de esta acción.	
El Sistema registra la baja del usuario (cliente) para la entidad activa	
Fin de CU	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Registrar Nuevo Usuario

Nombre del Use Case: Registrar Nuevo Usuario		Nro. de Orden: 15
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input checked="" type="checkbox"/> Complejo
	<input type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Usuario final	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Alta de nuevos usuarios finales del sistema		
Precondiciones:		
Post- Condiciones: El Sistema debe registrar los datos del nuevo usuario final del sistema		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario ingresa en la opción "Registrarse" de una página de rubro.		
El Sistema presenta un formulario de carga de datos para el nuevo usuario.		
El usuario ingresa los datos solicitados y presiona la opción "Aceptar"		
El sistema valida los datos ingresados y verifica que se hayan cargado correctamente todos los datos requeridos.	4.1.- El sistema comprueba que hay datos requeridos no ingresados o algún dato no ingresado correctamente. 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.	
5.- El Sistema registra el nuevo usuario		
6.- Fin de CU		
Use Case donde se incluye:		
Use Case al que extiende:		

Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

CU – Inicio de Sesión de Usuario

Nombre del Use Case: Iniciar Sesión de usuario		Nro. de Orden: 16
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Usuario	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Validar el ingreso del usuario al sitio web de rubro		
Precondiciones: El usuario debe estar registrado en el sistema		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario desea identificarse unívocamente en el sitio web e ingresa en "Iniciar Sesión"	1.1.- El Caso de Uso se inicia cuando una vez realizada la búsqueda de un turno deseado, el usuario desea realizar la reserva y aun no se encuentra logueado en el sitio.	
El Sistema presenta una pantalla donde solicita el ingreso de nombre de usuario (email) y contraseña.		
El usuario ingresa los datos y presiona "Aceptar"		
El Sistema verifica que la carga de datos es correcta.	4.1.- El sistema comprueba que hay datos ingresados incorrectamente 4.1.1.- El Sistema informa los errores. 4.1.2.- CU continúa en punto 3.	
El Sistema registra el ingreso del usuario en el sitio web.		
Fin de UC.		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Consultar Disponibilidad de Turnos

Nombre del Use Case: Consultar disponibilidad de turnos		Nro. de Orden: 17
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
	<input type="checkbox"/> Deseable	
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
	<input type="checkbox"/> Muy Simple	
Actor Principal: Usuario	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Buscar turnos disponibles entre los negocios aliados al proyecto		
Precondiciones: El usuario debe estar registrado en el sistema y debe haber iniciado sesión		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario ingresa en la pantalla de búsquedas de turnos en el sitio web.		
El Sistema presenta una pantalla con diferentes filtros de búsqueda.		
El usuario ingresa los datos en los filtros deseados y presiona "Aceptar"		
El Sistema busca los turnos disponibles de acuerdo a los filtros ingresados.		
El Sistema muestra la disponibilidad de turnos si es que hubiera.		
Fin de UC.		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Solicitar Turno

Nombre del Use Case: Solicitar turno		Nro. de Orden: 18
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Usuario	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Reservar turno en una entidad y espacio determinado		
Precondiciones: El usuario debe estar registrado en el sistema y debe haber iniciado sesión		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario desde la pantalla de "Consultar disponibilidad de turnos" y una vez ubicado el turno que desea reservar, ingresa al mismo para solicitarlo.		
El Sistema solicita la configuración de él o los canales de comunicación y pide confirmación de la reserva		
El Sistema registra la reserva, la cual según la configuración propia del espacio de la entidad para a la bandeja de entrada de pendientes de confirmación o bien, directamente a la agenda de turnos.		
Fin de UC.		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Administrar turnos solicitados

Nombre del Use Case: Administrar turnos solicitados		Nro. de Orden: 19
Prioridad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Usuario	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Mantener los turnos solicitados		
Precondiciones: El usuario debe estar registrado en el sistema, debe haber iniciado sesión y debe tener turnos solicitados		
Post- Condiciones:		
Curso Normal	Alternativas	
El Caso de Uso se inicia cuando un usuario selecciona la opción "Administrar turnos solicitados"		
El Sistema muestra un listado con todos los turnos solicitados y su estado actual, más la opción de "Cancelar" turno.		
El usuario selecciona un turno del listado y presiona la opción de Cancelar el turno.		
El Sistema verifica si es posible cancelar el turno y si es posible pide confirmación de la acción.		
El sistema registra la cancelación de turno		
Fin de UC.		
Use Case donde se incluye:		
Use Case al que extiende:		
Use Case de Generalización:		
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009	
Autor Última Modificación:	Fecha Última Modificación:	

CU – Configurar Datos Propios

Nombre del Use Case: Configurar Datos propios		Nro. de Orden: 20
Prioridad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Útil
		<input type="checkbox"/> Deseable
Complejidad:	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Complejo
	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Simple
		<input type="checkbox"/> Muy Simple
Actor Principal: Usuario	Actor Secundario:	
Tipo de Use Case:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Objetivo: Mantener los datos propios personales del usuario		
Precondiciones: El usuario debe estar registrado en el sistema y debe haber iniciado sesión.		

Post- Condiciones:	
Curso Normal	Alternativas
El Caso de Uso se inicia cuando un usuario selecciona la opción "Datos Generales"	
El Sistema muestra un formulario con los datos propios del usuario activo, con la posibilidad de editar determinados datos	
El usuario edita los datos que desee y presiona "Aceptar"	
El sistema registra los datos modificados del usuario.	
Fin de UC.	
Use Case donde se incluye:	
Use Case al que extiende:	
Use Case de Generalización:	
Autor: Bortagaray, Martín	Fecha Creación: 11/02/2009
Autor Última Modificación:	Fecha Última Modificación:

9.4 - Prototipo de Interfaz

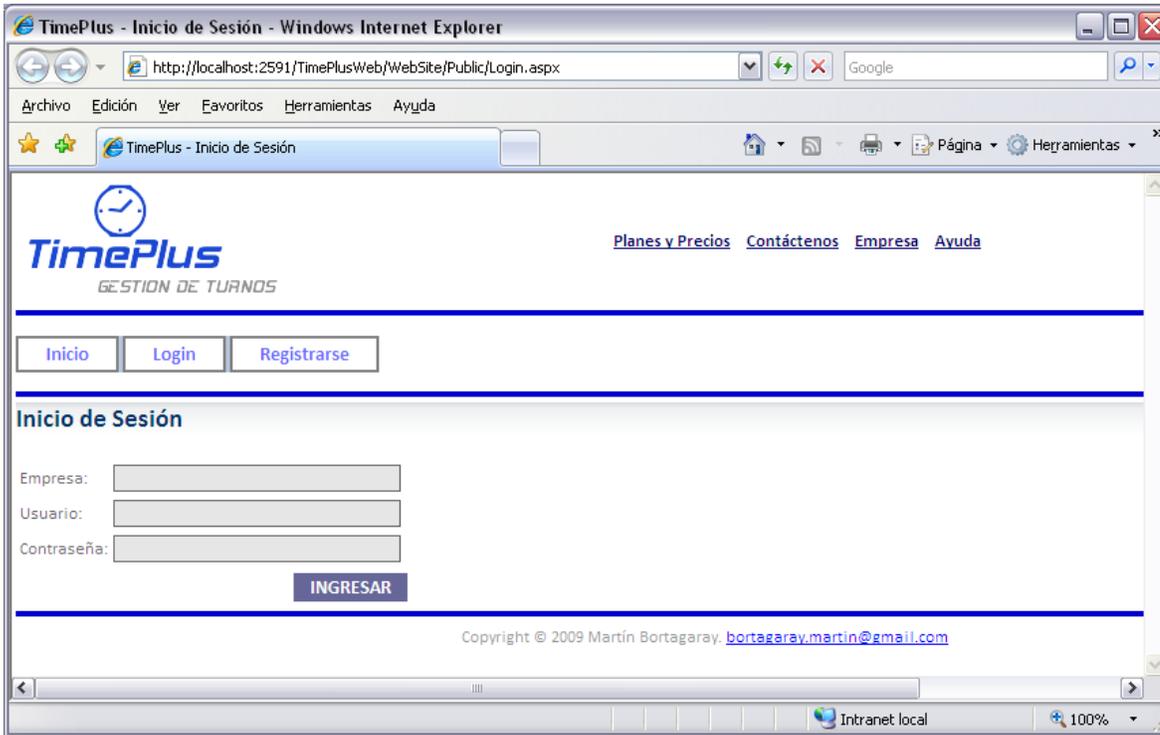
Se utilizará un técnica llamada **Prototipado funcional reutilizable**, que se basa en la creación de modelos o prototipos de interfaz que ayudan a evaluar la usabilidad de la aplicación web sin necesidad de esperar su implementación final. El término reutilizable se refiere a que no se pierde el esfuerzo efectuado en la construcción de prototipos, ya que estos fueron usados para construir el producto real.

Se exponen a continuación algunas pantallas representativas del sistema:

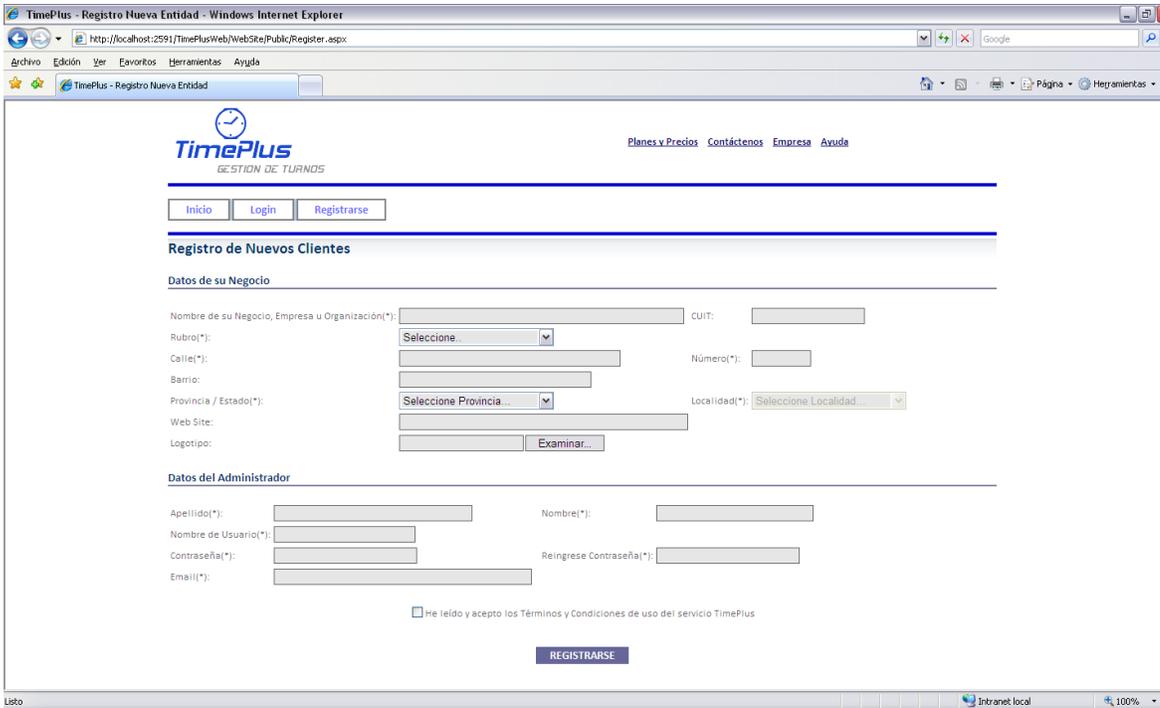
Pantalla Inicial

The screenshot shows the initial page of the TimePlus web application. The browser window is titled 'Untitled Page - Windows Internet Explorer' and the address bar shows 'http://localhost:2591/TimePlusWeb/WebSite/Public/Home.aspx'. The page features a blue header with the 'TimePlus' logo and the tagline 'GESTIÓN DE TURNOS'. A navigation menu includes links for 'Planes y Precios', 'Contáctenos', 'Empresa', and 'Ayuda'. Below the menu are buttons for 'Inicio', 'Login', and 'Registrarse'. The main content area is divided into several sections: 'INICIO DE SESIÓN' with fields for 'Empresa', 'Usuario', and 'Contraseña', and an 'INGRESAR' button; '¿QUÉ ES TIMEPLUS?' with a description of the system and a calendar image; '¿QUIÉNES PUEDEN USAR TIMEPLUS?' with a list of examples like 'Clubes Deportivos' and 'Consultorios Profesionales'; and 'CLIENTES' with a list of client types like 'Tennis Deportes' and 'Consultorios Privados'. There is also a 'REGÍSTRESE' section with three steps: 'Primero' (ingresar datos de empresa), 'Segundo' (ingresar datos de usuario), and 'Por último' (gestionar agendas). The footer contains the copyright notice: 'Copyright © 2009 Martín Bortagaray. bortagaray.martin@gmail.com'.

Pantalla de Inicio de Sesión



Pantalla de Registro de Nueva Entidad



Pantalla de Gestión de Turnos

TimePlus - Gestión de Turnos - Windows Internet Explorer

http://localhost:2591/TimePlusWeb/WebSite/App/Agenda.aspx

Tiene 2 solicitudes de turnos pendientes

Bienvenido Martín | Cerrar Sesión

TANCAT DEPORTES - Jacinto Rios 232 - B° Gral. Paz - Córdoba

Agenda Solicitudes Clientes Configuración Mi Cuenta

Gestión de Turnos

Categoría: Pádel

Hoy **miércoles, 26 de agosto de 2009** Día Semana Mes

	Cancha 1	Cancha 2	Cancha 3
09:00 a.m.			
10:00 a.m.			
11:00 a.m.			
12:00 p.m.			
01:00 p.m.			
02:00 p.m.			
03:00 p.m.			

Intranet local 100%

TimePlus - Gestión de Turnos - Windows Internet Explorer

http://localhost:2591/TimePlusWeb/WebSite/App/Agenda.aspx

Tiene 2 solicitudes de turnos pendientes

Bienvenido Martín | Cerrar Sesión

TANCAT DEPORTES - Jacinto Rios 232 - B° Gral. Paz - Córdoba

Agenda Solicitudes Clientes Configuración Mi Cuenta

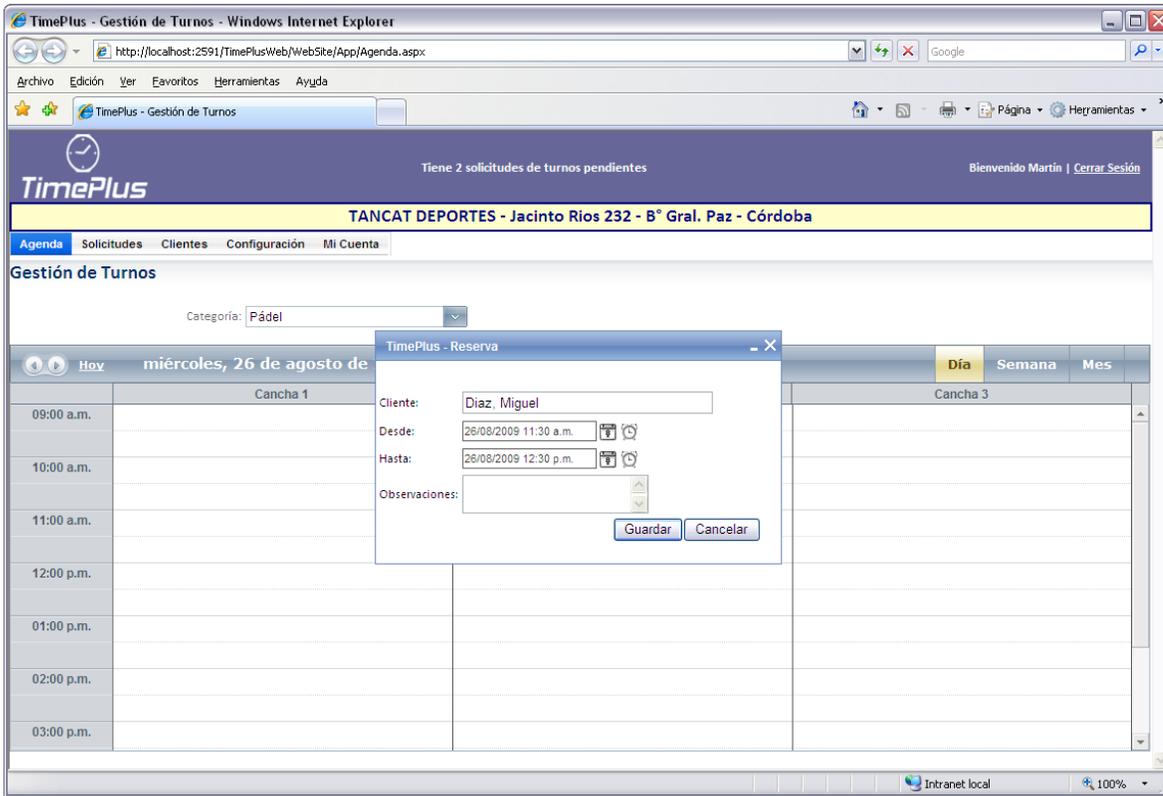
Gestión de Turnos

Categoría: Pádel

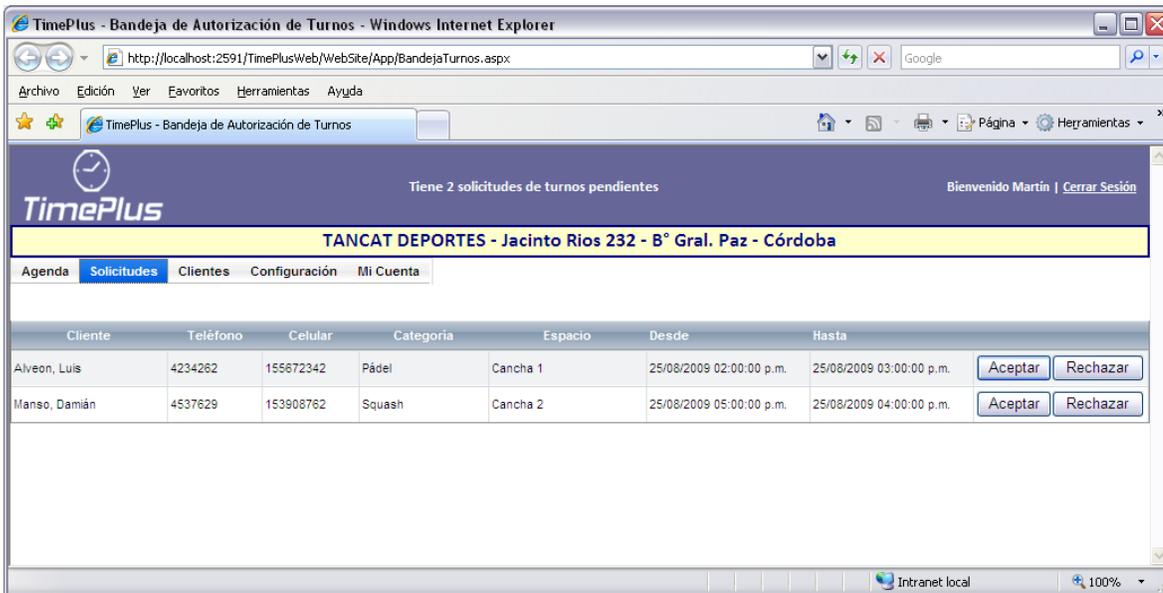
Hoy **23/08/2009 - 29/08/2009** Día Semana Mes

	Cancha 1							Cancha 2							Cancha 3						
	dom, 23	lun, 24	mar, 25	mié, 26	jue, 27	vie, 28	sáb, 29	dom, 23	lun, 24	mar, 25	mié, 26	jue, 27	vie, 28	sáb, 29	dom, 23	lun, 24	mar, 25	mié, 26	jue, 27	vie, 28	sáb, 29
09:00 a.m.																					
10:00 a.m.																					
11:00 a.m.																					
12:00 p.m.																					
01:00 p.m.																					
02:00 p.m.																					

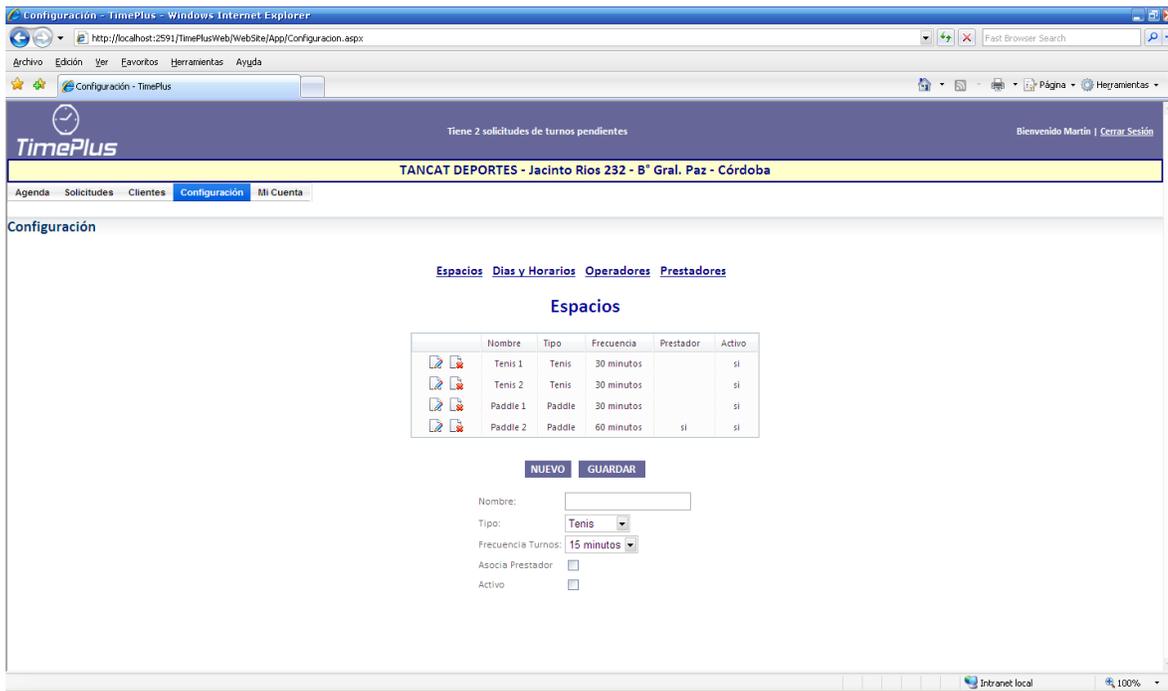
Intranet local 100%



Pantalla de Autorización de Turnos



Pantalla de Configuración (Espacios)



9.5.1 - Detalle de Diagrama de Entidad Relación

TABLA	DESCRIPCIÓN
AdmisionTipo	Contiene la modalidad de reserva por cada espacio, por ejemplo si se aceptan todos los usuarios o sólo los registrados en la entidad y si requiere confirmación de autorización de reserva por parte del negocio
Canales	Contiene los diversos canales mediante los cuales puede realizarse una reserva (Web, telefónico, presencial, y otros)
Dias	Contiene los días de la semana
DiasNoLaboral	Contiene día y horario no laborable por espacio (feriados, días sin atención)
Entidades	Contiene todas las organizaciones, empresas, negocios, y personas físicas que utilizarán la aplicación TimePlus para la gestión y administración de sus agendas de turnos para sus espacios de alquiler.
EntidadWS	Contiene la configuración del Web Service de las entidades que deseen utilizarlo para validar el ingreso de solicitudes de turnos
Espacios	Contiene los espacios de alquiler de las entidades, como ser, canchas de tenis, pádel, squash, fútbol, consultorios, talleres, habitaciones, mesas, entre otros.
EspaciosDiasHorarios	Contiene la configuración de días y horarios de disponibilidad para los espacios de una entidad
EspacioTipo	Contiene la tipificación de espacios, por ejemplo para el rubro deportes, cancha de fútbol, cancha de tenis, cancha de pádel, entre otros.
Estados	Contiene los estados de una reserva (pendiente de confirmación, confirmado, consumado)
Frecuencias	Contiene las fracciones en las que se dividirá una agenda de turnos de un espacio determinado, por ejemplo, 30 minutos, 1 hora, 2 horas, etc.
Horarios	Contiene el rango de hora desde, hora hasta que un espacio tendrá disponible para su agenda
Localidades	Contiene las localidades por provincia
Operadores	Contiene los operadores que utilizarán TimePlus para la administración y Gestión de turnos.
OperadoresEntidades	Contiene la relación entre Operadores y Entidades, es decir que operadores corresponden u operan en una entidad en particular
OperadorTipo	Contiene la tipificación para la tabla Operadores (Administrador de Sistema, Administrador Operador de Entidad, Operador de Entidad)
Países	Contiene los países donde pueden ubicarse negocios o entidades aliadas
ParametrosConfig	Contiene un XML de parámetros de configuración por si fuese necesario agregarlos a futuro para entidades y espacios
Prestadores	Contiene un listado de prestadores, que son las personas que prestan el servicio junto al espacio (si fuera necesario), por ejemplo, para un consultorio el prestador sería el kinesiólogo, médico, psicólogo, etc.
PrestadoresEntidad	Contiene la relación entre prestadores y Entidades, es decir que prestadores corresponden u operan en una entidad en particular
PrestadorTipo	Contiene la tipificación para prestadores (profesores, médicos, psicólogos, etc.)
Provincias	Contiene las provincias
Reservas	Contiene las reservas de turnos y las solicitudes de reservas de turnos para un espacio de una entidad
ReservaTipo	Contiene la tipificación de la reserva, por ejemplo si es una reserva común, interna o turno fijo, es decir con periodicidad.
Rubros	Contiene los rubros para las entidades, como ser deportes, consultorio, taller, etc.
Sesiones	Contiene un registro histórico de ingresos al sistema ya sean Operadores de TimePlus o usuario finales de un sitio web determinado
Unidades	Contiene las unidades para medir los tiempos (minutos, horas, días, etc.)
UsuarioEntidad	Contiene la relación entre usuarios finales y entidades
Usuarios	Contiene los usuarios finales del sistema, son quienes solicitarán y reservarán turnos en los espacios de las entidades

BIBLIOGRAFÍA

- David A. Ruble: Análisis y Diseño Práctico Para Sistemas Cliente / Servidor con GUI, México, Editorial Prentice Hall Latinoamericana; 1998.
- C. J. Date: Introducción a los Sistemas de Bases de Datos, México, 7ª Edición: Pearson Educación; 2001.
- Roger S. Pressman: Ingeniería del Software: Un Enfoque Práctico, 5ª Edición: McGraw Hill; 2002.

PÁGINAS WEB CONSULTADAS

- <http://www.lavozdelinterior.com.ar/default.asp?edicion=/07/10/31/> - Información sobre accesos a Internet – 23/03/2008
- [http://msdn2.microsoft.com/es-es/library/4w3ex9c2\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/4w3ex9c2(VS.80).aspx) - Información sobre ASP.NET – 17/08/2008
- <http://www.webestilo.com/aspnet/aspnet00.phtml> - información ASP.NET – 19/08/2008
- <http://www.subgurim.net/Articulos/asp-net-general/3/por-que-asp-net.aspx> - Comparación de lenguajes web - 10/09/2008
- <http://www.w3c.es/Divulgacion/Guiasbreves/ServiciosWeb> - Información sobre Servicios Web – 15/09/2008
- http://www.exo.com.ar/exos/paginas/plantillas_contenido/Page.asp?seccion=258&pagina=47&plantilla=Page.asp - Información sobre SQL Server 2005 – 22/09/2008

ANEXO 1: REFERENCIA DE HONORARIOS

Cargo	Por Mes	Por Hora
Administrador de ISP	\$1500	\$7 - \$9
Administrador de Redes	\$1500 - \$2500	\$20 - \$50
Analista Senior	\$1500 - \$2500	\$20 - \$50
Analista Junior	\$1000 - \$2000	\$20 - \$30
Analista Programador	\$1200 - \$2200	\$15 - \$30
Auditor Interno Informático	\$1800 - \$3500	
Consultor Informático		\$40 - \$70
Data Base Administrator	\$1500 - \$2500	
Data Entry (Por Dígito Alfanumérico)	\$0,002 - \$0,003	
Data Entry (Por Dígito Numérico)	\$0,001 - \$0,0015	
Diseñador Gráfico Senior	\$1100 - \$1800	\$20 - \$40
Diseñador Gráfico Junior	\$800 - \$1200	\$5 - \$8
Diseñador Industrial	\$1000 - \$1800	
Diseñador de Páginas Web	\$600 - \$900	\$4 - \$6
Dibujante con Articulación PC	\$800 - \$1500	
Forense	\$5000	\$100
Gerente de Sistemas PYMEs	\$2000 - \$3500	
Gerente de Sistemas de Grandes Empresas	\$4000 - \$5500	\$20 - \$100
Graboverificador		\$4 - \$8
Jefe de Servicio Técnico	\$900 - \$1400	
Jefe de Proyectos	\$1900 - \$2700	\$40 - \$70
Jefe de Gabinete Informático	\$1000 - \$1500	
Operador - Data Entry	\$600 - \$900	\$4 - \$6
Perito Informático		\$40 - \$70
Perito Operador PC	\$700 - \$950	
Programador de Páginas Web	\$800 - \$1200	\$5 - \$8
Programador Mayor	\$1100 - \$1900	\$7 - \$11
Programador Junior	\$800 - \$1200	\$5 - \$8
Profesor Informático	\$680 - \$850	\$15 - \$25
Instructor Informático		\$10 - \$20
Técnico de Hardware	\$600 - \$1000	\$4 - \$6

Fuente:

http://www.cpcipc.org.ar/portal/index.php?option=com_content&task=view&id=17&Itemid=33

ANEXO 2: ESTÁNDARES DE CODIFICACIÓN

Estándares de desarrollo

CÓDIGO FUENTE Y CONSTRUCCIÓN DE SOFTWARE

0. Introducción

Este documento provee convenciones, reglas y estilos para la creación de código fuente, y representa un resumen de los lineamientos requeridos para las actividades constructivas.

0.1 Alcance

Este documento alcanza a todos los elementos constructivos (código fuente, archivos y directorios, instaladores, etc.) del proceso de desarrollo de software.

0.2 Objetivo

Los estándares de codificación incrementan la productividad a través de dos mecanismos:

Los problemas de implementación son tratados uniformemente, evitando errores comunes, duplicación innecesaria y múltiples soluciones para el mismo problema.

El código fuente tiene formato y estilo consistentes, haciéndolo fácil de leer y comprender, incrementando la eficiencia y calidad, y reduciendo los costos de mantenimiento.

0.3 Background

Este documento está basado en el *CxOne Standard Source Code And Contruction*.

Muchas de las reglas en éste documento, así como en el CxOne Standard fueron tomadas del libro "*Code Complete*" de Steve McConnell.

0.4 Descripción

Este documento contiene un conjunto de reglas de construcción y codificación que son aplicables al programar en cualquier lenguaje y/o tecnología. Tanto como sea posible, todas las reglas y lineamientos deben ser utilizados en todos los lenguajes. Obviamente, esto no siempre es posible, y a veces las reglas específicas del lenguaje sobrescribirán explícitamente las reglas generales de éste estándar.

1. General

1.1 Mecánica

Si utiliza herramientas de generación de código que será extendido o mantenido, formatee el código de acuerdo a éste estándar de construcción.

Para crear un nuevo módulo o componente, comience con la plantilla apropiada. Si no existe ninguna plantilla apropiada, es preferible comenzar copiando un módulo o componente existente.

Iniciar desde cero cada módulo es ineficiente y puede introducir errores e inconsistencias de estilo.

Cuando utilice Copy/Paste, asegúrese de actualizar los comentarios.

1.2 Convención de nombres

Todos los nombres deben ser claros y descriptivos.

Esto incluye variables, archivos, funciones, clases, directorios, etc. No limitar artificialmente la longitud de nombres salvo que la tecnología lo demande.

1.3 Formato

Si no existe un estilo discernible en el código existente, formatee el código de acuerdo al estándar de construcción.

Si existe un estilo discernible pero difiere del estándar, considere reformatear el código de acuerdo al estándar.

Siga los ejemplos de formato en las plantillas de construcción.

Cuando no haya conflicto con el estándar de construcción, mantenga el estilo de documentos existentes.

Si está trabajando en un documento de otro desarrollador, mantenga un estilo consistente. No debe notarse que más de una persona trabajó sobre el documento.

En lo posible, mantenga la longitud de las líneas por debajo de los 80 caracteres.

Siempre indente el código. Para la indentación utilice TABS, no espacios.

En la mayoría de los entornos de desarrollo el indentado con TAB puede configurarse para mostrarse de mayor o menor tamaño. Configure éste valor en 4.

Use líneas libremente para separar y ordenar el código.

El código debe ser armónico para su lectura. Mantenga un balance visual que permita interpretar rápidamente el código.

En lo posible, inserte 1 línea blanca dentro de funciones, 2 líneas blancas para separar dos funciones juntas, y 3 o 4 líneas blancas para separar secciones dentro del archivo.

Coloque un espacio antes y después de cada operador (+, -, ==, etc.)

Ejemplo: `if (1 == 3)`

Coloque un espacio después de cada coma separando parámetros en la definición e invocación de una función.

Ejemplo: `funcionUno (int paramEntero, string paramString)`

Coloque un espacio antes y después de un índice de un array.

1.4 Comentarios

Utilice comentarios para describir la intención del programador, o proveer un resumen de una sección de código.

No use comentarios para escribir *que* hace el código, sino para explicar *porqué* lo hace.

Utilice frases completas en los comentarios, y mantenga la puntuación y capitalización de las palabras. No abrevie, ni resuma la información más de lo comprensible.

Indente los comentarios al mismo nivel de indentación que el código al cual se refieren.

Evite colocar comentarios de final de línea.

Ejemplo: `funcionUno (20); // Llamada a la función uno`

Utilice el marcador estándar `<IncompleteCode>` para indicar que una sección de código está incompleta y debe ser revisada.

Sintaxis del marcador de código incompleto:

`<IncompleteCode>Descripción </IncompleteCode>`

Ejemplo:

`<IncompleteCode>Falta validación de input</IncompleteCode>`

2. Módulos y archivos

Basar nuevos archivos de código fuente en las plantillas de construcción correspondientes.

Los nombres de archivos deben contener únicamente caracteres alfanuméricos.

Capitalizar la primer letra y la primer letra de cada palabra dentro del nombre, y usar minúscula para el resto de los caracteres.

Ejemplo: `AsientosContables.cs`

Inicie cada archivo con el comentario estándar para el encabezado, describiendo brevemente el propósito del archivo.

3. Variables

Utilice nombres de variables descriptivos. No utilice abreviaturas.

Un nombre de variable debe ser legible, preciso, describir su propósito sin ambigüedades. No use variables de un carácter salvo iteradores en ciclos for.

Usar minúscula para la primer letra del nombre de variable, capitalizar la primer letra de cada palabra dentro del nombre, y usar minúscula para el resto de los caracteres.

Ejemplo: `int balanceTrimestral;`

En lo posible, inicialice la variable en su declaración.

4. Constantes

Nunca utilice *números mágicos* dentro del código, aún cuando se utiliza una sola vez, siempre declare un nombre simbólico para una constante numérica.

5. Estructuras de control

Utilice nombres descriptivos para índices de loops en lo posible.

Si se está iterando sobre un array de peras, es preferible definir el índice como pera en lugar de i, o j.

Ejemplo:

```
for ( int pera = 0; pera < arrayDePeras.length; pera++ )
```

No utilice instrucciones goto.

6. Funciones

Usar minúscula para la primera letra del nombre de variable, capitalizar la primer letra de cada palabra dentro del nombre, y usar minúscula para el resto de los caracteres.

Ejemplo: emitirBalance(2004);

Anteceda la declaración de las funciones importantes con un comentario indicando su propósito.

Los nombres de parámetros de una función deben prefijarse con la palabra: param.

Ejemplo: funcionPrueba (int paramEntero)