

Especificación de Requisitos de Software

Enunciado del problema

Un centro de masajes experimenta problemas con la reserva de turnos. Actualmente, las reservas se preestablecen mediante llamadas, mensajes de WhatsApp o bien, personalmente. Una vez confirmada una reserva, la dueña del centro registra la misma en una agenda. Cancelaciones y cambios de último momento generan importantes problemas de organización para el negocio, hechos que repercuten negativamente de dos formas distintas: espacios de tiempo improductivos entre turno y turno en los que no se atiende a ningún cliente, y, en contraste, superposición de turnos en una misma franja horaria.

Solución propuesta

Se propone implementar el sistema *Carelly* para la reserva de turnos en el centro de masajes con el objetivo de mejorar la organización de los tiempos. El sistema permitirá al centro contar con información detallada de cada cliente. Además de información personal y de contacto, tendrá también un histórico de las veces que el cliente fue atendido en el centro.

Antes de registrar una nueva reserva (o modificar una existente) el sistema verificará los espacios de tiempo disponibles para la fecha especificada y los mostrará en pantalla. Una vez definido el horario para el cliente, el usuario podrá registrar la reserva (siempre y cuando no se solape con el de otra). Cada reserva puede tener asignada uno o más tipos de masajes, dependiendo del tipo de sesión que solicite el cliente. Se debe considerar que se puede dar de baja una reserva y su franja horaria debe quedar disponible para ser reutilizada por otra. Adicionalmente, el sistema permitirá listar clientes y sus respectivas reservas.

La característica principal del sistema será una vista en forma de cronograma, donde se mostrarán las reservas del turno para dicho día y/o para dicha semana. En este cronograma se detallarán los datos de cada reserva a fin de brindar información de utilidad a la dueña del centro. De esta manera, podrá tener una visión de los turnos por atender a lo largo distintas jornadas, como así también de los espacios de tiempo disponibles para asignar a clientes que deseen realizar una nueva reserva de turno (o bien, modificar una existente).

Clientes potenciales

Centro de masajes *Carelly*.

Requisitos funcionales

Deben estar:

- Como usuario, quiero registrar nuevos clientes para asociarlos a reservas de turnos.
- Como usuario, quiero gestionar la información de los clientes registrados para contar siempre con sus datos actualizados.
- Como usuario, quiero gestionar los distintos tipos de masajes que ofrece el centro para asociarlos a cada turno en particular.
- Como usuario, quiero gestionar las reservas de turno para tener mejor organizados los horarios del centro.

- Como usuario, quiero ver los detalles de cada turno para saber qué tipo de masaje se hará y durante cuánto tiempo se atenderá cada cliente.
- Como usuario, quiero tener un historial de turnos de cada cliente para saber si están siguiendo una rutina especial.
- Como usuario, quiero listar los turnos del día para saber a qué clientes atenderé durante la jornada.
- Como usuario, quiero ver un cronograma semanal para tener una visión completa de la semana.

Sería bueno que estén:

- Como usuario, quiero listar los clientes registrados para poder acceder a la información de alguno en particular.
- Como usuario, quiero ver una lista de los clientes que más reservas hicieron para ofrecerle descuentos o promociones.
- Como usuario, quiero ver estadísticas de los días y horarios que más reservas se hacen para conocer las preferencias de los clientes.

Arquitectura de software

Carelyly será una aplicación web que se enmarcará en la arquitectura Cliente-Servidor:

- Para el cliente (*frontend*), usaremos HTML, CSS y JavaScript. Es posible que implementemos el framework Bootstrap, que nos facilitará la creación de una interfaz de usuario moderna, elegante y receptiva.
- Para el servidor (*backend*), utilizaremos el lenguaje de programación Java en combinación con el framework Springboot.
- Implementaremos la base de datos PostgreSQL para proporcionar persistencia.