Programación Web 3

UNLaM - Tecnicatura en Desarrollo Web

Trabajo Práctico

1. Objetivo

Este documento describe el alcance funcional y los requisitos técnicos del trabajo práctico que los alumnos de la materia Programación Web III deberán aprobar a fin de estar en condición de aprobar/promocionar la materia..

2. Equipo

El equipo para realizar el trabajo práctico deberá ser de **4 alumnos, sin excepciones.**

3. Requisitos Técnicos

3.1 Proyecto .NET

1. El trabajo práctico deberá ser realizado utilizando ASP.NET. El tipo de proyecto a utilizar es una aplicación web.

3.2 Estilos

1. No se permitirán que se utilicen los estilos ya provistos por Microsoft en la aplicación de ejemplo que provee Visual Studio.

2. Todos los archivos .css deberán estar dentro de una carpeta.

3. No utilizar estilos inline (atributo style="") ni definir estilos dentro de una pagina (tags <style>).

4. Debe de utilizarse algún framework/biblioteca de hojas de estilo. Algunos ejemplos:
    a. Twitter Bootstrap (http://getbootstrap.com/ , temas http://bootswatch.com/)
    b. Foundation (http://foundation.zurb.com/docs/)

3.3 JavaScript

1. No utilizar JavaScript inline dentro de una página, se deberá referenciar a archivos js.

2. Todos los archivos .js deberán estar dentro de una carpeta.
    a. Si se decide utilizar algún js que no es propio, el mismo deberá estar dentro de una subcarpeta.

Recomendaciones

Las funciones específicas de una página, deberían estar en un archivo .js con el mismo nombre de la página
.
Aquellas funciones utilizadas en más de una página, deberían de estar dentro de otro archivo .js de uso común.

3.4 HTML

1. No utilizar tags table para organizar el contenido de una página en columnas, los tags

table solo están permitidos para representar una grilla/listado de información.

2. Todo el contenido debe ser SEO friendly.

3. Se requiere el uso de master page para estructura los formularios web de la aplicación. Dentro de la master page deberán referenciarse las hojas de estilo y archivos de javascript de uso común por toda la aplicación.

4. Debe utilizarse xHtml.

3.5 Validación

1. Utilizar validaciones tanto del lado del cliente (JavaScript) como del lado del servidor utilizando sólo controles ASP.NET.

2. Se puede utilizar una lista que detalle todos los campos que no cumplieron con las validaciones.

3. Se requerirá que las validaciones en los formularios webs serán realizadas mediante controles de validación de ASP.NET

3.6 Arquitectura y Consideraciones de Desarrollo

1. **IMPORTANTE:** Junto con el enunciado se les adjunta un .zip con la solución a utilizar.
    a. **No deben modificar los nombres de los archivos** existentes en el .zip.
    b. **No deben modificar los ID de los controles y dejarle ClientIDMode="Static"** en las páginas, pero pueden re-ordenarlos como quieran, deben agregarle clases para css y rodearlos con otros tags de html para que el diseño quede presentable.

2. La capa de acceso a datos deberá ser realizada con Entity Framework Este componente de .NET será explicado en clases a fin de que los alumnos comprendan cómo utilizarlo.

3. Utilizar la menor cantidad posible de código en los archivos aspx.cs, ascx.cs, master.cs, etc. e intentar que en los mismos haya llamadas a métodos dentro de otro proyecto que contenga las reglas de negocio.

4. Compatibilidad con exploradores.
    a. Internet Explorer 11, Edge
    b. Firefox (la última versión para Windows).
    c. Google Chrome (la última versión para Windows).

4. Objetivo del Proyecto

El objetivo del proyecto consiste en brindar un sitio web donde propietarios puedan publicar en alquiler espacios propios para ser utilizados como estacionamiento de vehículos.

Cada espacio podrá ser ofertado en diferentes períodos de fechas y en horarios determinados por el propietario. Por el alquiler de los espacios, los propietarios cobrarán un monto fijo por hora.

Los clientes podrán gestionar reservas de los espacios y podrán puntuar el lugar alquilado para que el resto de los clientes conozcan la reputación del lugar.

5. Especificación Funcional

Resumen de funcionalidad requerida:

1. Página de inicio (**/default.aspx**)
2. Menú específico para cada grupo
3. Registro de usuario (**/registracion.aspx**)
4. Login/Logout de usuario (para login usar **/login.aspx** y el logout puede ser un hyperlink o LinkButton)

Grupo Propietarios(es necesario estar logueado)

5. Crear cochera (**/propietarios/cocheras.aspx**).
6. Perfil (**/propietarios/perfil.aspx**)
7. Reservas (**/propietarios/reservas.aspx**)

Grupo Clientes(es necesario estar logueado)

8. Reservar cochera (**/clientes/reservar.aspx**)
9. Mis reservas (**/clientes/reservas.aspx**).
10. Puntuación (**/clientes/puntuacion.aspx**)

5.1 Pantalla Inicio

Cuando se accede al sitio web, los usuarios verán un buscador donde podrán indicar la ciudad en donde quieren reservar una cochera, la fecha de llegada y la fecha de salida del lugar. Como resultado de la búsqueda, los clientes verán la cantidad de resultados y todas las cocheras registradas con disponibilidad para ese período de fechas. Debe completar al menos uno de los 3 campos.

El buscador deberá ser desarrollado dentro de un **user control**. Este user control, también, será utilizado dentro del punto 5.8 del trabajo práctico.

Para verificar si tienen disponibilidad se deberá analizar si existen reservas de cada cochera en el período indicado. En caso de que haya reservas existentes la cochera no se deberá mostrar.

Para los casos, en los que el espacio tenga reservas ya finalizadas se deberá obtener cual es el promedio de la puntuación dada por los clientes y deberá ser mostrado en pantalla.

Por cada disponibilidad se deberá mostrar la siguiente información: precio, nombre y apellido del propietario, precio total por las horas que se desean reservar, la foto, el mapa del lugar donde está ubicado (utilizar la API de Google Maps) y la puntuación promedio

En el caso de no encontrar cocheras disponibles, mostrar un mensaje “No se encontraron resultados”.

Al hacer click, se le requerirá al usuario que se loguee/registre al sistema para luego ser enviado a la pantalla de reservar para la cochera seleccionada.

Se deberá enviar vía query string (usar /clientes/reservar.aspx?idcochera=123) el identificador de cuál fue el espacio seleccionado por el cliente. Una vez que el cliente, se lo loguee exitosamente, será redirigido a la pantalla para poder realizar la reserva del espacio seleccionado.

El desarrollo del método para obtener el listado de cocheras disponibles deberá ser realizado a través de un web service /servicios/Cocheras.asmx con el método **List<cocheraDTO> ObtenerCocheras(ubicacion, fechaInicio, fechaFin)** donde **cocheraDTO** debe ser una nueva clase con las propiedades necesarias a visualizar en el navegador.

5.2 Menú y Master Pages

Siempre habrá un menú visible.

Menú para Anónimos (sin estar logueado)

- Inicio (**/default.aspx**).
- Login (**/login.aspx**).
- Registración (/**registracion.aspx**).

Menú Propietarios

- Crear cochera (**/propietarios/cocheras.aspx**).
- Perfil (**/propietarios/perfil.aspx).**
- Reservas (**/propietarios/reservas.aspx**). **PÁGINA DE INICIO.**
- Logout

Menú Clientes

- Reservar cochera (**/clientes/reservar.aspx**).
- Mis reservas (**/clientes/reservas.aspx**). **PÁGINA DE INICIO.**
- Logout

Crear 4 master pages, Base.master, Anonimo.master, Comensal.master, Cocinero.master.

En **Base.master** se debe poner todo el html compartido (links a css, js, etc).

Anonimo.master debe implementar la master Base.master y será implementada por la página de Inicio y Login.

Propietarios.master debe implementar la master Base.master, será implementada por las páginas asociados al propietario. En el code behind debe validar que el usuario esté logueado, caso contrario redirigir a la página de Login, y que sea Propietario, caso contrario redirige a la página de Inicio de su perfil.

Clientes.master debe implementar la master Base.master y será implementada por las páginas asociados al cliente. En el code behind debe validar que el usuario esté logueado, caso contrario redirigir a la página de Login, y que sea Cliente, caso contrario redirige a la página de Inicio de su perfil.

5.3 Login/Logout.

La pantalla contendrá un campo email y contraseña para ingresar al sitio. Al ingresar en el menú del sitio estará visible un linkbutton de logout que al clickearlo borrará los datos de sesión y redirigirá a la página de Inicio.

En caso que el usuario que se loguea sea un perfil de propietario, el usuario debe ser redireccionado a la pantalla “Reservas”.

En caso que el usuario logueado sea un cliente debe ser redireccionado a la pantalla “Mis Reservas”.

Validación:

- **Email** - Requerido y Formato de email
- **Contraseña** - Requerido
- **Botón “Ingresar”,** en caso de que se haya accedido a la página de login a través de otra página, redirigir automáticamente, luego de hacer click en “Ingresar”
- Mostrar mensaje “Usuario y/o Contraseña inválidos”, en caso de que no se el usuario o la combinación de usuario y contraseña ingresada.

5.4 Registración.

En esta pantalla, un usuario puede registrarse en el sitio como uno de los perfiles posibles: propietario o cliente. Los datos a registrar son: nombre, apellido, email, contraseña, confirmación de contraseña y perfil (cliente o propietario). Este último campo deberá ser implementado con radios button donde el value es para cliente 1 y propietario 2.

Se debe validar todos los campos como obligatorios. Respecto a la contraseña debe estar ofuscado (tapado con caracteres). Debe cumplir con la condición de que tener como mínimo 8 caracteres, empezar con mayúscula y contener al menos 1 número (utilizar una expresión regular). Además, se debe validar que los campos contraseña y confirmación de contraseña sean iguales.

Además, se deberá validar que el email ya no se encuentre registrado en un usuario del sistema, en tal caso mostrar un mensaje “No se pudo completar la registración, existe otro usuario con el email ingresado.”

En caso de que la registración ocurra exitosamente, mostrar por pantalla “Registración exitosa, diríjase al login” (en login utilizar un link <a> que dirija a la página de login).

5.5 Crear cochera (Propietario).

El propietario puede crear cocheras para ser reservadas que luego son las se ofrecerán a los clientes. Para crear una cochera se deberá cargar la siguiente información:

- ubicación.
- período disponible (fecha inicio - fecha fin). Se deberá implementar con dos controles: uno para la fecha de inicio y otro para la fecha de fin.
- horario diario (hora inicio - hora fin). Se deberá implementar con dos controles. Cada uno deberá contener las horas con intervalos de 30 minutos.
- descripción de la cochera.
- latitud de ubicación (valor decimal)
- longitud de ubicación (valor decimal).
- metros cuadrados de la superficie (valor entero positivo).
- tipo de vehículo que se pueda estacionar (auto, pickup, camiones, moto). Se podrá hacer una selección múltiple de los tipos de vehículos que se pueden estacionar.
- precio por hora (valor decimal positivo).
- foto de la cochera.

Todos los campos son obligatorios y se deben validar los tipos de datos en los casos que corresponda.

Se deberán contemplar las siguientes validaciones:

- la fecha de inicio no puede ser mayor que la fecha de fin.
- la hora de inicio no puede ser mayor a la hora de fin.
- el precio de la cochera debe ser mayor a 0 (cero).
- los metros cuadrados no deben ser menores a 5.

Para subir la foto deben utilizar el control asp:UploadFile.
En caso de que la creación ocurra exitosamente, mostrar por pantalla “Operación exitosa”.

5.6 Perfil (Propietario).

El perfil permitirá que el propietario vea los datos con los cuales se registró dentro del sistema. Permitiendo editar sus datos a excepción del email. El email una vez registrado no se podrá modificar bajo ninguna circunstancia.

Se deberá validar que todos los datos estén completos.
En caso de que la actualización de información ocurra exitosamente, mostrar por pantalla “Operación exitosa”.

5.7 Reservas (Propietario).

El propietario podrá ver todas las reservas que se han registrado en sus cocheras. Las reservas podrán ser filtradas por período de fechas (desde - hasta) permitiendo ver períodos definidos. Los datos que se verán son:

- fecha de inicio.
- fecha de fin.
- ubicacion de cochera
- cantidad de horas.
- usuario que reservo.
- puntuación de la cochera.
- total cobrado.

El periodo de fechas a filtrar no puede ser mayor de 90 días. Se deberá validar y en caso que no se cumple notificar al usuario con un mensaje en pantalla “El periodo de fechas a filtrar no puede ser mayor de 90 días”.

Las reservas que aún no han sido efectivas (fechas futuras) deberán visualizarse con otro color que las distingan.

5.8 Reservar cochera (Cliente).

Un cliente podrá reservar una cochera registrada previamente en el sistema. Para poder encontrar la cochera deseada, el sistema le permitirá realizar una búsqueda dentro del sitio por nombre de ubicación (requerido), similar a la pantalla de inicio de la aplicación.

Recordar utilizar el user control planteado dentro del punto 5.1.

En el caso de no encontrar cocheras disponibles, mostrar un mensaje “No se encontraron resultados”.

Por cada disponibilidad se deberá mostrar la siguiente información: precio, nombre y apellido del propietario, precio por hora, la foto, el mapa del lugar donde está ubicado (utilizar la API de Google Maps) y la puntuación promedio.

Dentro de esta pantalla podrá seleccionar la cochera a reservar. Una vez seleccionada, el cliente será redirigido a una nueva pantalla de confirmación (/clientes/confirmar-reserva.aspx?idcochera=123) donde verá la siguiente información:

- fecha inicio
- fecha fin
- hora de entrada.
- hora de salida
- ubicación.
- imagen de la cochera
- precio por hora.
- precio total a abonar (solo lectura). Al cambiar alguno de estos campos: fecha inicio, fecha fin, hora entrada, hora salida y estando los 4 completos, calcular por javascript el precio total a abonar, una ayuda en pseudocódigo sería:
- dias = DiferenciaDeDias(fecha fin, fecha inicio)
- horas = DiferenciaDeHoras(hora salida, hora entrada)
- horas totales = horas * dias
- **precio total = horas totales * precio hora**

Ejemplo de como cambiar un span cuando cambia el valor en un input https://jsfiddle.net/hozx9eq6/5/

Debe ser tenido en cuenta que diferentes usuarios pueden reservar la misma cochera mientras no entren en conflicto, no puede haber 2 reservas que se solapen en dia y horario, pero si puede haber para una cochera N reservas en el mismo dia mientras sean en distintos horarios.
Ejemplo:

1. Dado la cochera “La cochera del Balon de Oro” que tiene disponibilidad del 17/10/16 hasta 24/10/16 entre las 08.00 - 18.00.
    a. Este es un caso posible:
        i. El usuario “Cristiano” reserva para el plazo 18/10/16 hasta 20/10/16 entre 08.00 - 10.00hs
        ii. El usuario “Lionel” reserva para el plazo 18/10/16 hasta 21/10/16 entre 12.00-18.00hs.
        iii. El usuario “Neymar” reserva para el plazo 20/10/16 hasta 24/10/16 entre las 10.00-12.00hs

Al confirmar la reserva será guardada en base de datos.
En caso de que la actualización de información ocurra exitosamente, mostrar por pantalla “Operación exitosa”.

La misma pantalla de confirmación deberá ser usada para cuando el clientes es redirigido desde la pantalla principal.

En caso de que el cliente tenga más de 2 reservas pendientes sin puntuar, no podrá realizar una nueva reserva hasta que les asigne puntuación.

5.9 Mis reservas (Cliente).

Los clientes tendrán una opción donde podrán ver todas las reservas que hayan realizadas ordenadas por fecha de inicio descendente Las reservas que ya han sido utilizadas (previas a la fecha actual) deberán estar en color gris.

Los datos que se deberán mostrar por cada reserva son:

- fecha inicio.
- fecha fin
- horario
- precio total.
- puntuación (en caso de que la reserva aún no tenga puntuación, se deberá mostrar la opción de puntuar (que abrirá el popup definido en el punto 5.10). En caso de que la reserva sea futura no se deberá mostrar ninguna opción.

5.10 Puntuación (Cliente).

El cliente podrá puntuar el espacio de estacionamiento reservado. Sólo se podrán ser puntuadas las reservas que ya hayan sido utilizadas y sólo podrá hacerlo por única vez. Una vez puntuada la reserva la puntuación se verá dentro de “Mis reservas” del cliente.

La opción de puntuación deberá estar en la misma pantalla del punto 5.9. No se deberá desarrollar en una nueva pantalla.

La puntuación consistirá en una valoración del 1 al 5. Utilizar un DropDownList.

Para cargar la puntuación se deberá desarrollar un pop-up dentro de la pantalla de “Mis Reservas” donde se permita seleccionar el valor de la puntuación.

En caso de que la puntuación ocurra exitosamente, actualizar el listado de mis reservas.

6. Forma de Entrega

El TP deberá ser enviado a los siguientes destinatarios, copiando a todos los integrantes del equipo en el email enviado.

- mpazwasiuchnik@unlam.edu.ar; pablokuko@gmail.com; pnsanchez@unlam.edu.ar;
- el resto de los integrantes que participaron del tp (como CC)

El email deberá contener los siguientes puntos:

a. El título del email será 2016-2C-TP-[Integrantes]
b. Respuestas a las siguientes preguntas:
    i. ¿Qué nota creen que deberían sacar en el tp? (1-10, donde para 7 debe estar toda la funcionalidad pedida) y ¿por qué?
    ii. ¿Qué cosas creen que podrían mejorarse?
    iii. ¿Qué les resultó más complicado?
c. Un Archivo [Grupo-integrante1-integrate2-etc].zip con toda la solución para que pueda ser compilada y ejecutada en cualquier computadora. **Cambiarle la extension al zip y que sea [Grupo-integrante1-integrate2-etc].txt, porque puede que el mail rebote por seguridad.**

PRIMER ENTREGA

Como para la **primer entrega no habrá Base de Datos**, deben considerar lo siguiente:
Los únicos usuarios validos serán:

- **cliente@gmail.com**, password "Password1" que será del tipo Cliente
- **propietario@gmail.com**,password "Password1" que será del tipo Propietario
- Si ingresa otro usuario o password mostrarán el mensaje correspondiente
- Luego de ingresar, si el usuario venía de otra pagina y fue redirigido acá por no estar logueado, redirigir a esa pagina, caso contrario redirigir a la página de inicio de ese tipo de usuario.

Los grupos deben ser de 4 personas.
Que deben desarrollar?

1. Agregar solo el maqueteado (html y controles de asp.net) para que cumpla con TODA la funcionalidad requerida.
2. **Les adjuntamos tambien un .zip con la solucion que deben utilizar,** por favor, **usen estas paginas, y usen los controles** que les pusimos y **no les cambien el ID**, pueden

cambiar el html que los rodea, cambiarlos de lugar, pero deben estar presentes en la página. Esto es necesario para nuestra corrección ya que ya están los casos de pruebas automatizados. Van a tener que agregar mas controles para cumplir con el enunciado, por ejemplo para mostrar listado de información (GridView, Repeater o DataList).

3. **Agregar TODAS las validaciones** que pide el enunciado (cliente y servidor), esto se verá próximamente en la clase de Controles de Validacion
4. **Agregar User Control requerido**
5. **Agregar los Redirects pedidos**
    a. Si el usuario desea acceder a una página que necesita estar logueado, guardar en session esta url, y redirigir al login, luego de que el usuario haya ingresado un login correcto (no hay Base de datos en la primer entrega), redirigir a la página deseada inicialmente.
    b. pueden guardar en sesión el tipo de usuario (cliente o propietario) y redirigirlo a donde corresponde.
    c. Si el usuario intenta acceder a una página que no le corresponde, por ejemplo propietario@gmail.com intenta acceder a una página dentro de /clientes/ deberá ser redirigido a su pagina de inicio (/propietarios/reservas.aspx), esta lógica deben agregarla en la Masterpage de Clientes.Master, lo mismo si un cliente intenta acceder a una página de propietario, y esta lógica la introducen en Propietarios.Master
6. **(Actualizado el dia 2016-09-23)** Alumn@s, como parte de la entrega del TP, deben enviar este EXCEL que les adjunto (correcciones-2016-2C.xlsx)marcando con una B cada item que tengan funcionando en la entrega, a la derecha verán si pueden rendir o no. Basado en el Porcentaje de Bien que tengan, estos son los posibles resultados:
    a. **X >=95% -->** "PUEDE RENDIR"
    b. **95% < X >=75% -->** "NO PUEDE RENDIR, DEBE REENTREGAR PARA RENDIR"
    c. **x < 75% -->** "NO PUEDE RENDIR, REENTREGA PARA RENDIR EL RECUPERATORIO"

7. Condiciones de Aprobación

1. El TP deberá cumplir con todos los requisitos técnicos y funcionales definidos. No se aceptará que alguna funcionalidad o requisito técnicos no se encuentren plasmados en el trabajo práctico.

2. Al momento de la entrega del trabajo práctico, todos los alumnos de cada equipo deberán estar presentes para la defensa del trabajo práctico.

3. En la defensa del trabajo, se evaluará el grado de conocimiento y participación en el desarrollo del trabajo práctico de cada alumno.