****

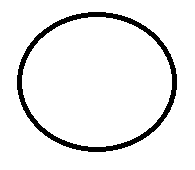
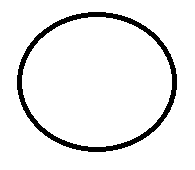
**OBLIGATORIO**

Programación 1

**“*Gestión de Consultorio Médico*”**

**Ver.28/06**

**Entrega:28/06**



|  |
| --- |
| **Integrantes:** Mesa Gonzalez, Alvaro Nro.187251  Bruno Fleischer Nro.\*\*\*\*\*\*\*\*  **Carrera:** Analista en Tecnologías del a Información.  **Grupo:** N1D.  **Docente:** Alejandro Martins |

**Indice**

1)Planteo del Problema.

1.1)Aplicación para Gestión de Consultorio Médico. Pg.3

1.2)Sobre los Reportes. Pg.4

2)Selección de Tecnologias para el desarrollo.

2.1)Criterios de la solución. Pg.6

2.2)Requerimientos de la solución.

3)Gestion de Perfiles.

3.1)Perfiles de la Aplicación. Pg.8

3.2)Perfil del usuario de la Aplicación.

3.3)Perfil del paciente.

3.4)Perfil del médico.

4)Desarrollo de los Reportes.

4.1)Tabla de datos del Reporte.

5)Resumen de la aplicación Web.

5.1)Listado de Variables.

5.2)Listado de Funciones.

6)Presentaciónes de la Aplicación.

6.1)Diseño de la Aplicación.

6.2)Usabilidad de la Aplicación.

7)Pruebas Funcionales de la Aplicación.

7.1)Funcionalidad de los Usuarios.

7.2)Funcionalidad de los Datos precargados.

7.3)Funcionalidad de las Consultas.

7.4)Funcionalidad del Paciente.

7.5)Funcionalidad del Médico.

7.6)Funcionalidad de los formularios.

7.7)Funcionalidad del Reporte.

8)Codigo Fuente de la Aplicación.

8.1)HTML.

8.2)Librerias.

8.3)Objetos.

9)FAQ.

10)Anexo.

10.1)Usuarios y claves para login.

**Planteo del Problema.**

**1.1)Aplicación para Gestión de Consultorio Médico.**

**Objetivo:**

* Desarrollar un prototipo de aplicación para la gestión de un consultorio médico.
* Se incluirá la documentación pertinente.

**Proceso de desarrollo de la aplicación.**

La aplicación debe estar desarrollada con HTML, JS,CSS e imágenes utilizando jQuery como framework para el desarrollo.

**Gestión de perfiles.**

La aplicación contará con dos perfiles de usuarios, el perfil paciente y el perfil médico, cada uno tendrá acceso a diferentes funcionalidades dentro de la aplicación.

**Perfil paciente.**

El paciente podrá:

* Solicitar una consulta para un médico en particular.En este punto no se debe hacer manejo de agenda ni horarios, el usuario simplemente elige el medico con el cual hacer la consulta y se crea la consulta nueva.La consulta tendrá un número único identificador que estará compuesto por la primera letra del nombre del médico, las primeras tres letras del apellido del médico y un valor numérico único autoincremental.
* Modificar sus datos personales.
* Ver sus consultas finalizadas y la descripción.
* Ver sus consultas no finalizadas.
* Todas las opciones anteriores estarán disponibles solamente cuando el usuario esté habilitado, salvo la opción de modificación de datos personales que estará disponible en todo momento para el usuario.

**Perfil médico.**

El médico podrá:

* Realizar Consulta.(Atender al paciente).Se despliegan las consultas del médico que no hayan sido finalizadas, incluyendo el nombre del paciente.

A continuación podrá anteder a ese paciente utilizando las siguientes opciones:

* Asignar peso y altura de un paciente.Con esta información se deberá:
* Almacenar esa información como dato del paciente; no se debe llevar un histórico de peso y altura, pero sí se deberá informar al doctor si el peso ingresado es el nuevo máximo o nuevo mínimo histórico del paciente.
* Calcular su índice de masa corporal verificando, si respecto a su última consulta el IMC tuvo una variación de más del 10% en este caso se debe informar al doctor de la situación.
* La consulta tendrá, la referencia del paciente al que corresponde, el medico solicitante,si fue paga o no y si está finalizada o no.
* Ver las consultas impagas del paciente y marcarlas como pagas.
* Finalizar Consulta: le agregará un texto descriptivo.La consulta quedará marcada como finalizada.
* Habilitar o inhabilitar pacientes. Se ingresara su cédula, se muestra el estado actual y se permite habilitar/inhabilitar según corresponda.

**De los doctores se conoce:** su nombre, apellido, su especialidad, número de profesional.Cada médico tiene una sola especialidad.

**De los pacientes se conoce:** su nombre, apellido, su número de paciente y una foto.

Además de estos datos se podrán incorporar todos los datos que el alumno entienda necesarios, por ejemplo manejo de contraseñas para cada tipo de usuario.

**Reportes.**

El sistema deberá realizar un listado en tabular.

Se incluirá: Nombre del Médico, su Especialidad y la cantidad de consultas finalizadas.

El listado estará siempre visible para todos los usuarios y se actualizará automáticamente.

Estará ordenado ascendentemente por Especialidad. Dentro de una misma especialdad estará ordenado descendentemente por cantidad de consultas del médico.

|  |  |  |
| --- | --- | --- |
| **Especialidad** | **Nombre** | **Cantidad de Consultas** |
| Cardiología | Pedro Gómez | 40 |
| Cardiología | María Perez | 30 |
| Oftalmología | Rodrigo Torena | 25 |
| Urología | Mauricio González | 6 |

Se solicita:

* Precargar información de usuarios y datos en el sistema, para poder probar las funcionalidades sin tener que cargar datos cada vez que se inicia la aplicación.Se deben precargar al menos 10 médicos y 10 pacientes, procurando utilizar variedad en los datos, por ejemplo, asignando consultas a distintos médicos de la misma especialidad.
* Hacer las validaciones de datos que se entiendan necesarias para el correcto funcionamiento de la aplicación y la prevención de errores.
* Crear todas las funcionalidades que permitan dar respuesta a los requerimientos planteados.

Se valorará:

* La optimización del código de la solución, procurando no repetir código y logrando dividir problemas en partes pequeñas.
* El uso de nombres de variables y funciones mnemotécnicas.

**Producto**

El producto final deberá ser:

* La aplicación web que se deberá poder ejecutar sin inconvenientes a partir del CD entregado.
* La documentación, que consistirá en un archivo PDF (único) que deberá contener lo siguiente:
* Portada de obligatorio con nombre y foto de cada uno de los integrantes del grupo.
* Índice con los capítulos de la documentación, y la página en la que se puede consultar.
* Explicación de las variables y funciones javascript/jquery utilizadas (creadas por los integrantes del equipo). Para las variables se indicará su nombre, tipo, ámbito y su objetivo (qué almacenan).
* El código completo (html y js) que deberá incluir comentarios.
* Listado de información precargada para médicos, y pacientes. Incluir todos los datos.

Para este punto es importante considerar los estándares propuestos por la Facultad (doc. 302) al momento de confeccionar la documentación. Los mismos están publicados en aulas en el grupo bedeliá.

Tomar en cuenta que esos estándares están establecidos para la presentación de proyectos finales de carrera, por lo que se tomarán como guía.

**Estructura de carpetas de la entrega.**

El CD deberá tener dos carpetas, una para la documentación del proyecto y otra con la aplicación funcionando.

**Cronograma de hitos (avances) del obligatorio.**

Cada hito consistirá en una muestra parcial del avance del obligatorio al docente. Las fechas y mecanismo de la presentación de avances serán comunicadas en clase por cada docente.El cronograma se publicará en aulas oportunamente.

**Entregar en Bedelía:**

Un sobre tamaño A5 que contenga dos CD idénticos.

* Un CD **original** y otro de **respaldo**  conteniendo todos los archivos y carpetas de la aplicación solicitada.Los Cds estarán rotulados con el nombre y número de estudiante de los integrantes del equipo del obligatorio. La documentación no se entrega impresa, solamente en formato digital. **No olvide inscribirse. Las boletas de inscripción firmadas por los integrantes del equipo obligatorio deberán entregarse juntamente con el CD en bedelía.**

**Respaldo en aulas:**

Subir en aulas en la tarea correspondiente un respaldo de la aplicación y la documentación en un único archivo comprimido (RAR/ZIP).

**Esta tarea en aulas NO SUSTITUYE la entrega formal bedelía.**

**La entrega en bedelía es la oficial y es obligatoria, de no hacerlo se perderán los puntos del obligatorio.**

**Defensa:**

* La fecha de la defensa se coodinará oportunamente con cada docente.
* La asistencia a la defensa es **OBLIGATORIA.** No asistir a la defensa implicará la perdida de todos los puntos del obligatorio.

**Importante:**Todas las consultas y/o modificaciones que puedan surgir para el desarrollo del obligatorio serán evacuadas a través del foro "Consultas de obligatorio" disponible en aulas. Las aclaraciones, enmiendas, etc. A la letra que publiquen los docentes a través del foro se considerarán parte integral de la letra, por lo que será responsabilidad del estudiante tomarlas en cuenta para la entrega cuando corresponda.

Las librerías a utilizar son las planteadas en la letra, en caso de querer utilizar otros recursos se deberá consultar al docente con antelación mínima de 15 días a la fecha de entrega para evaluar su viabilidad.

Los ejercicios son individuales, dos aulmnos no pueden plantear una solución igual o similar para los mismos, en el caso de que esto sucediera el valor de la entrega será 0 (cero).

El alumno es el responsable por la integridad de los datos provistos.

**Selección de Tecnologias para el desarrollo.**

**2.1)Criterios de la solución.**

Nuestra aplicación estara desarrollada mediante el uso de las siguientes tecnologias:

**HTML5, CSS3 (bootstrap 3) y Javascript ES5 (jQuery).**

<!-- Comentario :

Quizas en esta parte resumir que tecnologias vamos a usar y si podemos me encargo de explicar un resumen de cada tecnologia, capaz para que la documentación tenga más peso.

-->

**Librerias y Frameworks:**

A su vez se aplicara para el diseño de la aplicación el framework Bootstrap 3 para un diseño adaptativo y de rápida implementación , el cual nos permitira proveer de una aplicación que pueda usar el cliente sin importar del dispositivo ( el cual se adaptara a la pantalla tanto de pc, tablet o smartphone).

Para el desarrollo de la aplicación de gestión se utilizara la libreria jQuery y jQuery UI para facilitar la implementación de la programación dinamica.

**2.2)Requerimientos de la solución.**

**Gestion de Perfiles.**

**3.1)Perfiles de la Aplicación.**

**3.2)Perfil del usuario de la Aplicacion.**

Constructor del objeto cliente:

//Creamos un constructor para el objeto cliente.

/\*al ser un constructor, su nomenclatura comienza en Mayuscula, lleva de argumentos

las propiedades del objeto que ya conocemos nombre, apellido, su foto, numero de paciente. \*/

//con el constructor creamos objetos que utilizaran tanto los perfiles de clientes y medicos.

function Usuario(nombre, apellido, cedula,usuario,contraseña)

{

/\* propiedades del objeto usuario tienen como valores su nombre apellido y la opcion de elegir usuario y contraseña \*/

this.nombre = nombre;

this.apellido = apellido;

this.nombreCompleto = nombre + " " + apellido;

this.cedula = cedula;

this.id = nombre +""+ cedula;

this.usuario = usuario || this.nombreCompleto;

this.contraseña = contraseña || cedula;

// los metodos que le permiten modificar sus propiedades.

this.modificarNombre = function(nuevoNombre){

this.nombre = nuevoNombre;

};

this.modificarApellido = function(nuevoApellido){

this.apellido= nuevoApellido;

};

this.modificarCedula = function(nuevaCedula){

this.cedula= nuevaCedula;

};

this.modificarUsuario = function(nuevoUsuario){

this.usuario = nuevoUsuario;

};

this.modificarContraseña = function(nuevaContraseña){

this.contraseña= nuevaContraseña;

};

};

Este constructor tendra las propiedades y metodos que todo los usuarios.

De esta manera el usuario al ingresar a la aplicación recibirá un formulario pidiéndole esta información.

Estas propiedades y metodos seran heredadas a los objetos Cliente y Medico, donde cada uno tendra sos propiedades unicas, pero seran accesibles.

**3.3)Perfil del cliente.**

Constructor del objeto cliente:

//Creamos un constructor para el objeto cliente.

/\*al ser un constructor, su nomenclatura comienza en Mayuscula, lleva de argumentos

las propiedades del objeto que ya conocemos nombre, apellido, su foto, numero de paciente. \*/

//Heredar las propiedades y metodos del constructor Usuario.

Cliente.prototype = new Usuario;

function Cliente(numeroPaciente, foto)

{

//El usuario ingresa su numero de paciente de la policlinca y una foto.

this.numeroPaciente = numeroPaciente;

this.foto = foto || ‘./images/default.jpg’;

this.modificarFoto = function(nuevaFoto){

this.foto = nuevaFoto;

}

};//cierra constructor Cliente;

/\*detallar otras propiedades y metodos\*/

//Como prueba se cargan 10 clientes.

var test = {};

test.x = 0;

test.clients = [

new Cliente(112, './images/user112.jpg'),

new Cliente(141, './images/user141.jpg'),

new Cliente(665, './images/user665.jpg'),

new Cliente(406, './images/user406.jpg'),

new Cliente(605, './images/user605.jpg'),

new Cliente(520, './images/user520.jpg')

];

//Cargar en consola los doctores disponibles al comenzar el turno.

console.table(test.clients);

**3.4)Perfil del médico.**

Constructor del objeto medico:

//Creamos un constructor para el objeto medico

//De los doctores se conoce

/\*al ser un constructor, su nomenclatura comienza en Mayuscula, lleva de argumentos

las propiedades nombre, apellido, su especialidad, numero de profesional. \*/

//Heredar las propiedades y metodos del constructor Usuario.

Medico.prototype = new Usuario;

function Medico(numeroProfesional, especialidad)

{

//El usuario ingresa su numero de paciente de la policlinca y una foto.

this. numeroProfesional = numeroProfesional;

this. especialidad = especialidad ;

};//cierra constructor Cliente;

/\*detallar otras propiedades y metodos\*/

//Como prueba se cargan 10 doctores.

test.y= 0;

test.doctors = [

new Medico("Oftalmología",107),

new Medico("Pediatra",227),

new Medico("Oftalmología",307),

new Medico("Cirujano",207),

new Medico("Oftalmología",507),

new Medico("Pediatra",347),

new Medico("Cirujano",527),

new Medico("Oftalmología",337),

new Medico("Psiquiatra",157),

new Medico("Pediatra",527)

];

//Cargar en consola los doctores disponibles al comenzar el turno.

console.table(test.doctors);

**Desarrollo de los Reportes.**

**Resumen de la aplicación Web.**

**Listado de Variables:**

/\* Estructuras de datos de mayor importancia aqui \*/

**Listado de Funciones:**

**Funciones resumidas y ordenadas por archivo :**

/\*

Los metodos a crear el usuario:

* function para levantar datos de formulario y validar.
* Function para generar numero id unico de usuario.

Los Metodos a crear para el paciente:

* function para que cliente Genere Consulta;
* function para que Ciente Modifique sus Datos personales(nombre,apellido y foto);

(Disponible en todo momento);

* function para Ver Consultas Finalizadas y descripcion;

(Disponible solo habilitado);

* function para Ver Consultas Abiertas;

(Disponible solo habilitado);

Los Metodos a crear para el medico:

* function para Atender Consulta;
* function para Asignar Peso y Altura de un paciente;
* Almacenar registro de ultimo maximo o minimo;
* function para Calcular IMC del paciente;
* function para Verificar Variacion de mas del 10%, informar al medico;
* function para Corroborar si Consulta fue paga o no, finalizada o no;
* function para Ver consultas impagas del pacientes y Marcarlas como pagas;
* function para Finalizar consulta, agregando un texto descriptivo, dejándola como finalizada;

Los metodos para las consultas y reportes:

* function para realizar presentación tabular de los reportes;
* Ordenar por especialidad, dentro de misma especialidad descendentemente por cantidad de consultas del medico;
* function para crear Numero Identificador de la consulta,  
  (compuesto por la primera letra del nombre del medico, las primeras tres letras del apellido del medio y un valor numerico unico autoincremental;

\*/

**Presentaciones de la Aplicación.**

**6.1)Diseño de la Aplicación.**

Una interfaz mas sencilla para ingresar como Cliente o Medico.

**Codigo Fuente de la Aplicación.**

**Presentar aquí el codigo fuente para leer;**

**FAQ**

**Frequently Asked Questions:**

**ANEXO.**

**Usuario y Claves de Pacientes**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Usuario** | **Clave** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Usuario y Claves de Pacientes**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Usuario** | **Clave** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |