

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 1



Ana Lía López – 290918



Andrés Scheck – 262086

[N1E]

Docente: [Bruno Díaz]

[Analista en Tecnologías de la Información]

[Fecha de entrega del documento (23-06-2022)]

Índice

1.	Descripción general del problema a resolver	5
1.1.	Tipos de usuario del sistema	5
1.2.	Listado de funcionalidades	5
2.	Detalle de Funcionalidades.....	7
2.1	F01 – Elección de tipo de usuario.....	7
2.1.1	Acceso	7
2.1.2	Descripción.....	7
2.1.3	Interfaz de usuario	7
2.1.4	Validaciones	7
2.2	F02 – Cerrar Sesión	7
2.2.1	Acceso	7
2.2.2	Descripción.....	7
2.2.3	Interfaz de usuario	8
2.2.4	Validaciones	8
2.3	F03 – Ingresar como Local	8
2.3.1	Acceso	8
2.3.2	Descripción.....	8
2.3.3	Interfaz de usuario	8
2.3.4	Validaciones	8
2.4	F04 – Seleccionar opción como Local.....	9
2.4.1	Acceso	9
2.4.2	Descripción.....	9
2.4.3	Interfaz de usuario	9
2.4.4	Validaciones	9
2.5	F05 – Habilitar y deshabilitar disponibilidad.....	10
2.5.1	Acceso	10
2.5.2	Descripción.....	10
2.5.3	Interfaz de usuario	10
2.5.4	Validaciones	10
2.6	F06 – Cambiar las reservas de estado	10
2.6.1	Acceso	10
2.6.2	Descripción.....	10
2.6.3	Interfaz de usuario	10

2.6.4	Validaciones	11
2.7	F07 – Modificar cupo máximo	11
2.7.1	Acceso	11
2.7.2	Descripción.....	11
2.7.3	Interfaz de usuario	11
2.7.4	Validaciones	12
2.8	F08 – Visualizar información estadística.....	12
2.8.1	Acceso	12
2.8.2	Descripción.....	12
2.8.3	Interfaz de usuario	12
2.8.4	Validaciones	12
2.9	F09 – Registrarse como Persona.....	12
2.9.1	Acceso	12
2.9.2	Descripción.....	13
2.9.3	Interfaz de usuario	13
2.9.4	Validaciones	13
2.10	F10 – Ingresar como Persona	13
2.10.1	Acceso	13
2.10.1	Descripción.....	14
2.10.2	Interfaz de usuario	14
2.10.3	Validaciones	14
2.11	F11 – Seleccionar opción como Persona.....	14
2.11.1	Acceso	14
2.11.2	Descripción.....	14
2.11.3	Interfaz de usuario	15
2.11.4	Validaciones	15
2.12	F12 – Solicitar una reserva.....	15
2.12.1	Acceso	15
2.12.2	Descripción.....	15
2.12.3	Interfaz de usuario	15
2.12.4	Validaciones	15
2.13	F13 – Cancelar una reserva	16
2.13.1	Acceso	16
2.13.2	Descripción.....	16
2.13.3	Interfaz de usuario	16
2.13.4	Validaciones	16

2.14 F14 – Calificar una reserva finalizada	16
2.14.1 Acceso	16
2.14.2 Descripción	16
2.14.3 Interfaz de usuario	16
2.14.4 Validaciones	16
2.15 F15 – Ver listado de reservas pendientes	17
2.15.1 Acceso	17
2.15.2 Descripción	17
2.15.3 Interfaz de usuario	17
2.15.4 Validaciones	17
2.16 F16 – Ver información estadística	17
2.16.1 Acceso	17
2.16.2 Descripción	17
2.16.3 Interfaz de usuario	17
2.16.4 Validaciones	17
3. Informe de Testing	18
3.1 FUNCIONALIDAD F10 - Ingresar como Persona	18
3.2 FUNCIONALIDAD F06- Cambiar Reservas de Estado	20
4. Lista de datos precargados	22
4.1 Usuarios Persona:	22
4.2 Usuarios Locales:	23
4.3 Reservas:	25
5. Código	30

1. Descripción general del problema a resolver

Se programará una aplicación donde usuarios podrán gestionar reservas. La aplicación contará con dos tipos de perfiles: perfil Persona, el cual podrá realizar reservas para asistir a locales como restaurants, teatros o museos, y perfil Local, el cual podrá habilitar o deshabilitar dichos eventos entre otras funcionalidades.

1.1. Tipos de usuario del sistema

- Persona
- Local

1.2. Listado de funcionalidades

F01 – Elección de tipo de usuario – Usuarios: Persona, Local

F02 - Cerrar Sesión - Usuarios: Persona, Local

F03 – Ingresar como Local – Usuario: Local

F04 – Seleccionar opción como Local – Usuario: Local

F05 – Habilitar y deshabilitar disponibilidad – Usuario: Local

F06 – Cambiar las reservas de estado – Usuario: Local

F07 – Modificar cupo máximo – Usuario: Local

F08 – Visualizar información estadística – Usuario: Local

F09 – Registrarse como Persona – Usuario: Persona

F10 – Ingresar como Persona – Usuario: Persona

F11 – Seleccionar opción como Persona – Usuario: Persona

F12 – Solicitar una reserva – Usuario: Persona

F13 – Cancelar una reserva – Usuario: Persona

F14 – Calificar una reserva finalizada – Usuario: Persona

F15 – Ver listado de reservas pendientes – Usuario: Persona

F16 – Visualizar información estadística – Usuario: Persona

2. Detalle de Funcionalidades

A continuación, se presenta el detalle de cada una de las funcionalidades a resolver en el obligatorio. Las funcionalidades están ordenadas por tipo de usuario.

2.1 F01 – Elección de tipo de usuario

2.1.1 Acceso

Podrán acceder los usuarios tipo Persona y Local.

2.1.2 Descripción

Permite elegir el tipo de usuario que va a ingresar a la aplicación.

2.1.3 Interfaz de usuario



BIENVENIDOS AL SISTEMA
DE RESERVAS

Seleccione tipo de usuario al que desea ingresar:

PERSONA

LOCAL

2.1.4 Validaciones

Es obligatorio elegir una opción para pasar a la siguiente pantalla. No hay validaciones. No existe posibilidad de error ya que solo debe elegirse una de las dos opciones posibles.

2.2 F02 – Cerrar Sesión

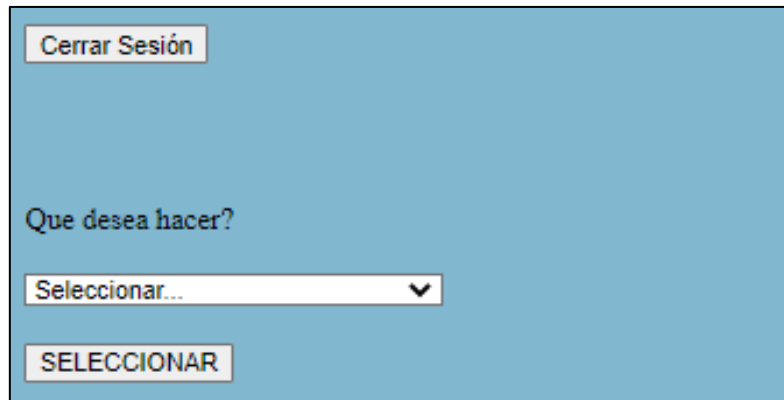
2.2.1 Acceso

Podrán acceder los usuarios tipo Persona y Local.

2.2.2 Descripción

Permite cerrar la sesión del usuario y volver a la página de inicio donde le permite volver a seleccionar los usuarios tipo Persona y Local. Se podrá cerrar sesión en cualquier momento.

2.2.3 Interfaz de usuario



A screenshot of a user interface with a light blue background. At the top left is a button labeled "Cerrar Sesión". Below it, the text "Que desea hacer?" is displayed. Underneath is a dropdown menu with the text "Seleccionar..." and a downward arrow. At the bottom is a button labeled "SELECCIONAR".

2.2.4 Validaciones

No hay validaciones. La funcionalidad solo estará visible si hay una sesión abierta.

2.3 F03 – Ingresar como Local

2.3.1 Acceso

Podrán acceder los usuarios tipo Local.

2.3.2 Descripción

Permite a los usuarios tipo Local ingresar a la aplicación.

2.3.3 Interfaz de usuario



A screenshot of a login form titled "USUARIO LOCAL:" in large, bold, black letters. Below the title is the instruction "COMPLETE LOS SIGUIENTES DATOS PARA INGRESAR". The form contains three input fields: "Usuario:" followed by a text box, "Contraseña:" followed by a text box, and "Seleccione tipo de local" followed by a dropdown menu showing "Restaurant" with a downward arrow. At the bottom is a button labeled "INGRESAR".

2.3.4 Validaciones

Los usuarios y las credenciales de acceso deberán estar precargados en la aplicación, por lo que se deberá validar contra esos datos.

El resto de la información (dirección, cupo máximo, foto y estado habilitado o no) estará precargada.

Se realizarán las siguientes validaciones particulares con sus correspondientes mensajes de error.

Campo	Validación	Mensaje de Error
-	Todos los campos son obligatorios	“Rellenar todos los campos”
Usuario	Debe estar precargado y ser único. Case insensitive.	“El usuario no existe”
Contraseña	Debe estar precargado. Case sensitive.	“Contraseña incorrecta, vuelva a intentar”
Tipo de Local	Debe estar precargado.	“Usted no cuenta con el tipo de local seleccionado”

2.4 F04 – Seleccionar opción como Local

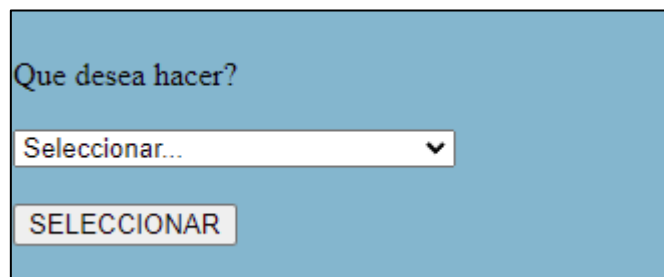
2.4.1 Acceso

Podrán acceder los usuarios tipo Local.

2.4.2 Descripción

Permite a los usuarios tipo Local seleccionar una opción de lo que desea hacer en la aplicación.

2.4.3 Interfaz de usuario

La imagen muestra una interfaz de usuario con un fondo azul. En la parte superior, hay un texto "Que desea hacer?". Debajo de este, hay un campo de selección (select) con el texto "Seleccionar..." y una flecha hacia abajo. En la parte inferior, hay un botón rectangular con el texto "SELECCIONAR" en mayúsculas.

2.4.4 Validaciones

La opción del select es obligatoria, en caso de no seleccionarla se mostrará el siguiente mensaje de error: “Debe seleccionar una opcion”.

2.5 F05 – Habilitar y deshabilitar disponibilidad

2.5.1 Acceso

Podrán acceder los usuarios tipo Local.

2.5.2 Descripción

Permite habilitar la posibilidad de que los usuarios tipo Persona puedan hacer reservas o deshabilitar la posibilidad de reservas temporalmente. Se deshabilitará automáticamente si la cantidad de reservas totales pendientes coincide con el cupo máximo del local.

2.5.3 Interfaz de usuario

Local	Estado de Disponibilidad	Acción
Teatro Solis	Habilitado	Habilitar/Deshabilitar

Una vez que se haga click en el boton “Habilitar/Deshabilitar”, se actualizará la pantalla. El Estado de Disponibilidad pasará a mostrar “Habilitado” o “Deshabilitado”.

2.5.4 Validaciones

En caso de que la cantidad de reservas totales pendientes coincida con el cupo máximo del local, se deshabilitará automáticamente y se mostrará un mensaje del tipo “Se ha deshabilitado la posibilidad de reservas” y se deshabilitará el botón de “Habilitar/Deshabilitar”.

2.6 F06 – Cambiar las reservas de estado

2.6.1 Acceso

Podrán acceder los usuarios tipo Local.

2.6.2 Descripción

Permite cambiar las reservas del local de un estado original “pendiente” a “finalizada”. Además, deberá contar con un buscador para buscar una reserva por nombre de cliente y poder marcarla como finalizada individualmente (solamente buscará en las reservas que estén pendientes).

2.6.3 Interfaz de usuario

Local	Persona	Cupos reservados	Estado de Reserva	Acción
Teatro Solis	Luis	5	Pendiente	Pendiente/Finalizada

Una vez que se haga click en el boton “Pendiente/Finalizada”, se actualizará la pantalla. El Estado de Reserva pasará a mostrar “Pendiente” o “Finalizada”.

Un formulario de búsqueda con un fondo azul claro. A la izquierda, el texto "Buscar Reserva:" en negro. A la derecha, un campo de texto blanco rectangular y un botón rectangular con el texto "Buscar" en negro.

Con respecto al buscador, cuando se haga click en el botón “Buscar”, se desplegara una tabla con la información mostrada arriba de la reserva buscada.

2.6.4 Validaciones

Para el botón de “Pendiente/Finalizada” no existen validaciones dado que se trata de un botón.

El campo de texto para la búsqueda es obligatorio. En caso de que el campo esté vacío, se mostrará el siguiente mensaje de error: “Debe ingresar un texto”.

En caso de que el campo de texto no esté vacío, aparecerán debajo en una tabla las reservas (buscadas por nombre de cliente) que están en estado “Pendiente”. Luego se tendrá la opción de hacer click en el botón “Pendiente/Finalizada” que aparecerá junto a cada reserva.

Si no existen coincidencias para el texto ingresado, se mostrará el siguiente mensaje de error: “No hay resultados que coincidan con el texto ingresado”.

2.7 F07 – Modificar cupo máximo

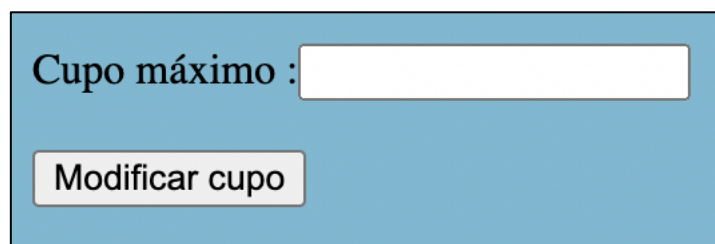
2.7.1 Acceso

Podrán acceder los usuarios tipo Local.

2.7.2 Descripción

Permite modificar el cupo máximo del local solamente si no se tienen reservas en estado pendiente.

2.7.3 Interfaz de usuario

Una interfaz de usuario con un fondo azul claro. En la parte superior, el texto "Cupo máximo :" en negro, seguido de un campo de texto blanco rectangular. En la parte inferior, un botón rectangular con el texto "Modificar cupo" en negro.

2.7.4 Validaciones

El campo de texto para la búsqueda es obligatorio. En caso de que el campo esté vacío, se mostrará el siguiente mensaje de error: “Debe ingresar un texto”.

En caso de que el local al cual se le quiere modificar el cupo máximo tengo reservas en estado pendiente, se mostrará el siguiente mensaje de error: “El local cuenta con reservas en estado pendiente”.

2.8 F08 – Visualizar información estadística

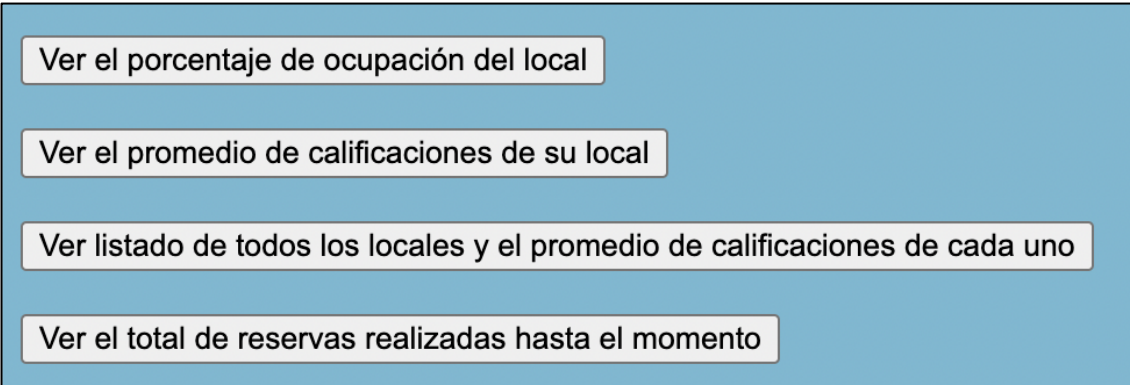
2.8.1 Acceso

Podrán acceder los usuarios tipo Local.

2.8.2 Descripción

Permite ver el porcentaje de ocupación del local, el promedio de calificaciones de su local junto a un listado de todos los locales y el promedio de calificaciones de cada uno, y el total de reservas realizadas hasta el momento.

2.8.3 Interfaz de usuario



Ver el porcentaje de ocupación del local

Ver el promedio de calificaciones de su local

Ver listado de todos los locales y el promedio de calificaciones de cada uno

Ver el total de reservas realizadas hasta el momento

2.8.4 Validaciones

La funcionalidad no requiere validaciones dado que solo se trata de botones que muestran información.

2.9 F09 – Registrarse como Persona

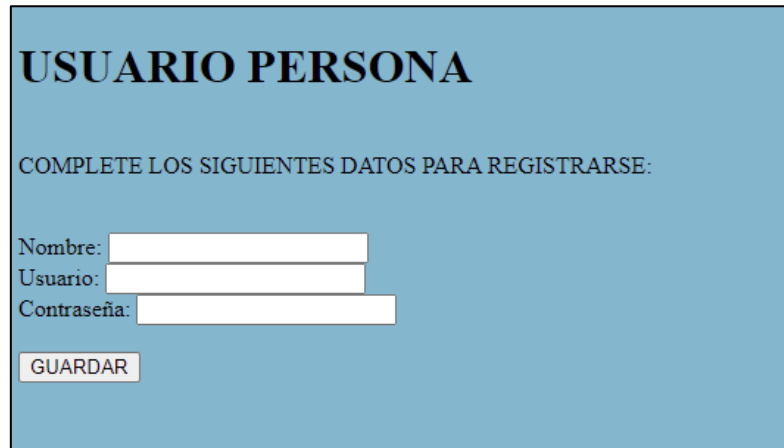
2.9.1 Acceso

Podrá acceder a esta funcionalidad cualquier usuario no registrado que desee registrarse como Persona.

2.9.2 Descripción

Permite al usuario registrarse como Persona y acceder a las funcionalidades de la aplicación.

2.9.3 Interfaz de usuario



USUARIO PERSONA

COMPLETE LOS SIGUIENTES DATOS PARA REGISTRARSE:

Nombre:

Usuario:

Contraseña:

2.9.4 Validaciones

Campo	Validación	Mensaje de Error
-	Todos los campos son obligatorios.	“Rellenar todos los campos”
Nombre	No se podrá repetir en dos usuarios diferentes en toda la aplicación y tampoco podrá coincidir con el nombre de usuario de un local.	“Nombre ya existe”
Usuario	No podrá coincidir con ningún otro usuario ya registrado.	“El usuario ya existe”
Contraseña	Deberá contener al menos 6 caracteres, entre los cuales deberá haber al menos 1 mayúscula, 1 minúscula y 1 número.	“Contraseña inválida. Debe contener al menos 6 caracteres e incluir 1 mayúscula, 1 minúscula y 1 número.”

2.10 F10 – Ingresar como Persona

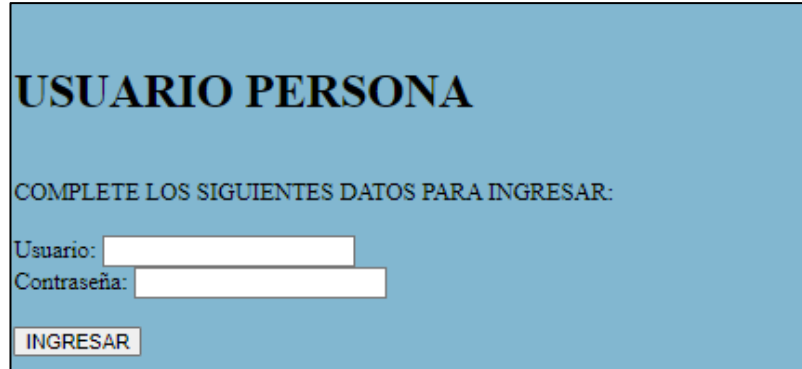
2.10.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.10.1 Descripción

Permite a los usuarios tipo Persona ingresar a la aplicación.

2.10.2 Interfaz de usuario



USUARIO PERSONA

COMPLETE LOS SIGUIENTES DATOS PARA INGRESAR:

Usuario:

Contraseña:

2.10.3 Validaciones

Se realizarán las siguientes validaciones particulares con sus correspondientes mensajes de error.

Campo	Validación	Mensaje de Error
-	Todos los campos son obligatorios	“Rellenar todos los campos”
Usuario	Debe estar registrado en la aplicación. Case insensitive.	“El usuario no existe o el tipo de usuario no coincide”
Contraseña	Debe estar asociada al usuario. Case sensitive.	“Contraseña incorrecta, vuelva a intentar”

2.11 F11 – Seleccionar opción como Persona

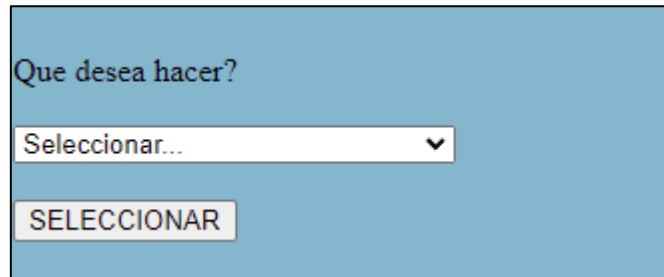
2.11.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.11.2 Descripción

Permite a los usuarios tipo Persona seleccionar una opción de lo que desea hacer en la aplicación.

2.11.3 Interfaz de usuario



Que desea hacer?

Seleccionar... ▼

SELECCIONAR

2.11.4 Validaciones

La opción del select es obligatoria, en caso de no seleccionarla se mostrará el siguiente mensaje de error: “Debe seleccionar una opcion”.

2.12 F12 – Solicitar una reserva

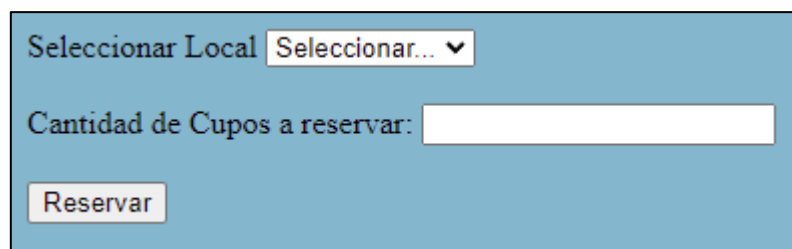
2.12.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.12.2 Descripción

Permite a los usuarios tipo Persona realizar la solicitud de una reserva mientras el usuario no tenga una reserva pendiente en ese local.

2.12.3 Interfaz de usuario



Seleccionar Local Seleccionar... ▼

Cantidad de Cupos a reservar:

Reservar

2.12.4 Validaciones

El campo de texto y la opción del select son obligatorios, además se validará que la cantidad de cupos a reservar sea mayor a cero. En caso de que no se seleccione un local o el campo de texto este vacío o la cantidad de cupos sea menor a cero, se mostrará el siguiente mensaje de error: “Debe seleccionar un local e ingresar al menos un cupo a reservar”.

En caso de que la validación anterior sea correcta, se solicitará la reserva si y solo si el usuario no tiene ya una reserva pendiente en ese local, teniendo en cuenta además las restricciones que se presentan por parte del usuario tipo Local.

2.13 F13 – Cancelar una reserva

2.13.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.13.2 Descripción

Permite a los usuarios tipo Persona cancelar una reserva “pendiente” propia pasando la misma a estado “cancelada”.

2.13.3 Interfaz de usuario

Local	Estado de Reserva	Acción
Teatro Solis	Pendiente	<input type="button" value="Cancelar"/>

Una vez que se haga click en el botón “Cancelar”, se actualizará la pantalla. El Estado de Reserva pasará a mostrar “Cancelada”.

2.13.4 Validaciones

No se requieren validaciones dado que la funcionalidad solo trata de un botón.

2.14 F14 – Calificar una reserva finalizada

2.14.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.14.2 Descripción

Permite a los usuarios tipo Persona ver un listado con sus reservas finalizadas y podrá puntuar esa reserva con un valor numérico de 1 a 5.

2.14.3 Interfaz de usuario

Reserva	Calificacion	Calificar
Teatro Solis	<input type="text" value="Seleccionar..."/>	<input type="button" value="Calificar"/>

2.14.4 Validaciones

La opción del select es obligatoria, en caso de no seleccionarla se mostrará el siguiente mensaje de error: “Debe seleccionar una calificación”.

2.15 F15 – Ver listado de reservas pendientes

2.15.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.15.2 Descripción

Permite a los usuarios tipo Persona ver un listado de sus reservas pendientes, se mostrará foto del local, nombre y cantidad de cupos.

2.15.3 Interfaz de usuario

Foto del Local	Nombre	Cantidad de Cupos
Foto	Teatro Solis	5

2.15.4 Validaciones

En caso de no tener reservas pendientes se mostrara el siguiente mensaje : “No cuenta con reservas pendientes”.

2.16 F16 – Ver información estadística

2.16.1 Acceso

Podrán acceder los usuarios tipo Persona.

2.16.2 Descripción

Permite a los usuarios tipo Persona ver un listado donde aparezcan los locales donde el usuario haya hecho reservas que haya completado, la cantidad de reservas que hizo, el total de reservas que tiene ese local y el porcentaje de reservas que corresponden a ese usuario en ese local. También permitirá al usuario ver el local en el que ha hecho más reservas.

2.16.3 Interfaz de usuario

Ver el porcentaje de reservas
Ver local en el que ha hecho más reservas

2.16.4 Validaciones

La funcionalidad no requiere validaciones dado que solo se trata de botones que muestran información.

3. Informe de Testing

3.1 FUNCIONALIDAD F10 - Ingresar como Persona

Nro	Prueba Realizada	Resultado deseado	Resultado obtenido	Resultado deseado y obtenido
1	Probar ingresar sin rellenar uno o varios campos del Login Persona.	No se logra ingresar a la aplicación. Se muestra mensaje de error: "Se deben completar todos los campos".	No se logra ingresar a la aplicación. Se muestra mensaje de error: "Se deben completar todos los campos".	TRUE
2	Probar ingresar con usuario no registrado.	No se logra ingresar a la aplicación. Se muestra mensaje de error: "El usuario no existe".	No se logra ingresar a la aplicación. Se muestra mensaje de error: "El usuario no existe".	TRUE
3	Probar ingresar con usuario existente, pero contraseña incorrecta.	No se logra ingresar a la aplicación. Se muestra mensaje de error: "Contraseña incorrecta, intente nuevamente".	No se logra ingresar a la aplicación. Se muestra mensaje de error: "Contraseña incorrecta, intente nuevamente".	TRUE
4	Probar ingresar con todos los datos correctos de un usuario registrado y además ingresando el nombre de usuario en minúscula (case insensitive).	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	TRUE
5	Probar ingresar con todos los datos correctos de un usuario registrado y además ingresando el	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	TRUE

	nombre de usuario en mayúscula (case insensitive).			
6	Probar ingresar con datos correctos de usuario precargado.	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	Se permite el ingreso y se muestra la pantalla de ya iniciado sesión de usuario Persona logueado.	TRUE

3.2 FUNCIONALIDAD F06- Cambiar Reservas de Estado

Nro	Prueba Realizada	Resultado deseado	Resultado obtenido	Resultado deseado y obtenido
1	Pantalla de Inicio de Sesión de Usuario Local: Click el botón [Cambiar las reservas de estado]	Se muestra pantalla con campo para buscar reserva por nombre de usuario y tabla con reservas pendientes para finalizar.	Se muestra pantalla con campo para buscar reserva por nombre de usuario y tabla con reservas pendientes para finalizar.	TRUE
2	Presionar el botón [Buscar nombre de usuario] con el campo vacío.	Se borran los datos de reservas pendientes de la tabla y se muestra mensaje de error: "Debe ingresar un texto".	Se borran los datos de reservas pendientes de la tabla y se muestra mensaje de error: "Debe ingresar un texto".	TRUE
3	Luego de presionar el botón [Buscar nombre de usuario] con el campo vacío, oprimir el botón [Reestablecer Tabla].	Se muestran todos los datos de reservas pendientes nuevamente en la tabla.	Se muestran todos los datos de reservas pendientes nuevamente en la tabla.	TRUE
4	Presionar el botón [Buscar nombre de usuario] con un texto que no se encuentre incluido en los nombres de las personas en la tabla de reservas pendientes.	Se muestra mensaje de error: " No se encontro usuario con ese nombre que tenga una reserva pendiente".	Se muestra mensaje de error: " No se encontro usuario con ese nombre que tenga una reserva pendiente".	TRUE
5	Presionar el botón [Buscar nombre de usuario] con un	Se borran los datos ingresados en el campo de texto y se muestran los datos de	Se borran los datos ingresados en el campo de texto y se muestran los datos de	TRUE

	texto que se encuentre incluido en los nombres de las personas en la tabla de reservas pendientes.	reservas pendientes de los nombres de usuario que incluyen el texto buscado en la tabla.	reservas pendientes de los nombres de usuario que incluyen el texto buscado en la tabla.	
6	Presionar el botón [Finalizar] de una de las reservas que aparece en la tabla de reservas pendientes.	La reserva cambio su estado de “pendiente” a “finalizada” y la misma se borra de la tabla de reservas pendientes.	La reserva cambio su estado de “pendiente” a “finalizada” y la misma se borra de la tabla de reservas pendientes.	TRUE
7	Presionar el botón [Buscar nombre de usuario] con el campo vacío y apretar botón [Volver]. Luego volver a presionar botón de [Cambiar las reservas de estado]	Se elimina el mensaje de texto "Debe ingresar un texto" y aparece vacío. Se muestran todos los datos de reservas pendientes en la tabla.	Se elimina el mensaje de texto "Debe ingresar un texto" y aparece vacío. Se muestran todos los datos de reservas pendientes en la tabla.	TRUE
8	Presionar el botón [Buscar nombre de usuario] con el campo vacío y apretar botón [Cerrar Sesión]. Luego volver a ingresar a la pantalla de Cambiar las reservas de estado	Se elimina el mensaje "Debe ingresar un texto".y aparece vacío. Se muestran todos los datos de reservas pendientes en la tabla.	Se elimina el mensaje "Debe ingresar un texto".y aparece vacío. Se muestran todos los datos de reservas pendientes en la tabla.	TRUE

4. Lista de datos precargados

4.1 Usuarios Persona:

1. nombre: Ana
nombreUsuario: UP1
password: An1234
idPersona: 1
2. nombre: Emiliano
nombreUsuario: UP2
password: Em1234
idPersona: 2
3. nombre: Juan
nombreUsuario: UP3
password: Ju1234
idPersona: 3
4. nombre: Andres
nombreUsuario: UP4
password: An1234
idPersona: 4
5. nombre: Florencia
nombreUsuario: UP5
password: Fl1234
idPersona: 5
6. nombre: Bruno
nombreUsuario: UP6
password: Br1234
idPersona: 6
7. nombre: Luis
nombreUsuario: UP7
password: Lu1234
idPersona: 7

4.2 Usuarios Locales:

1. nombre: Teatro Solis
nombreUsuario: UL1
password: Ts1234
tipoDeLocal: Teatro
direccion: Reconquista S/N
cupoMaximo: 1500
foto: teatroSolis.jpg
habilitado: true
idLocal: 1 (se genera automáticamente cuando se precarga el usuarioLocal)

2. nombre: El Galpon
nombreUsuario: UL2
password: Eg1234
tipoDeLocal: Teatro
direccion: Av. 18 de julio 1618
cupoMaximo: 800
foto: El-galpon.jpg
habilitado: true
idLocal: 2 (se genera automáticamente cuando se precarga el usuarioLocal)

3. nombre: Museo Juan Manuel Blanes
nombreUsuario: UL3
password: Mj1234
tipoDeLocal: Museo
direccion: Millán 4015
cupoMaximo: 50
foto: Sala_blanes.jpg
habilitado: true
idLocal: 3 (se genera automáticamente cuando se precarga el usuarioLocal)

4. nombre: Museo Artes Visuales
nombreUsuario: UL4
password: Mv1234
tipoDeLocal: Museo
direccion: Av. Tomas Garibaldi 2283
cupoMaximo: 100
foto: museo_artes_visuales.jpg

habilitado: true
idLocal: 4 (se genera automáticamente cuando se precarga el usuarioLocal)

5. nombre: Pantado
nombreUsuario: UL5
password: Rp1234
tipoDeLocal: Restaurante
direccion: Rbla. República del Perú
cupoMaximo: 100
foto: restaurante-plantado.jpg
habilitado: true
idLocal: 5 (se genera automáticamente cuando se precarga el usuarioLocal)

6. nombre: Garcia
nombreUsuario: UL6
password: Rg1234
tipoDeLocal: Restaurante
direccion: Av. Alfredo Arocena 1587
cupoMaximo: 100
foto: Garcia.jpg
habilitado: true
idLocal: 6 (se genera automáticamente cuando se precarga el usuarioLocal)

7. nombre: Café Misterio
nombreUsuario: UL7
password: Cm1234
tipoDeLocal: Restaurante
direccion: Costa Rica 1700
cupoMaximo: 50
foto: café_misterio.jpg
habilitado: true
idLocal: 7 (se genera automáticamente cuando se precarga el usuarioLocal)

4.3 Reservas:

1. estado: pendiente
cupos: 5
calificacion: null
idReserva: 1 (se genera automáticamente cuando se precarga la reserva)
idLocal: 1
idPersona: 1

2. estado: pendiente
cupos: 4
calificacion: null
idReserva: 2 (se genera automáticamente cuando se precarga la reserva)
idLocal: 2
idPersona: 1

3. estado: pendiente
cupos: 6
calificacion: null
idReserva: 3 (se genera automáticamente cuando se precarga la reserva)
idLocal: 2
idPersona: 2

4. estado: pendiente
cupos: 5
calificacion: null
idReserva: 4 (se genera automáticamente cuando se precarga la reserva)
idLocal: 4
idPersona: 2

5. estado: pendiente
cupos: 10
calificacion: null
idReserva: 5 (se genera automáticamente cuando se precarga la reserva)
idLocal: 3

idPersona: 3

6. estado: pendiente
cupos: 5
calificacion: null
idReserva: 6 (se genera automáticamente cuando se precarga la reserva)
idLocal: 5
idPersona: 3
7. estado: pendiente
cupos: 2
calificacion: null
idReserva: 7 (se genera automáticamente cuando se precarga la reserva)
idLocal: 4
idPersona: 4
8. estado: pendiente
cupos: 3
calificacion: null
idReserva: 8 (se genera automáticamente cuando se precarga la reserva)
idLocal: 7
idPersona: 4
9. estado: pendiente
cupos: 5
calificacion: null
idReserva: 9 (se genera automáticamente cuando se precarga la reserva)
idLocal: 6
idPersona: 4

10. estado: pendiente
cupos: 7
calificacion: null
idReserva: 10 (se genera automáticamente cuando se precarga la reserva)
idLocal: 5
idPersona: 5

11. estado: pendiente
cupos: 5
calificacion: null
idReserva: 11 (se genera automáticamente cuando se precarga la reserva)
idLocal: 2
idPersona: 5

12. estado: pendiente
cupos: 5
calificacion: null
idReserva: 12 (se genera automáticamente cuando se precarga la reserva)
idLocal: 1
idPersona: 5

13. estado: finalizada
cupos: 4
calificacion: 3
idReserva: 13 (se genera automáticamente cuando se precarga la reserva)
idLocal: 6
idPersona: 1

14. estado: finalizada
cupos: 4
calificacion: 3
idReserva: 14 (se genera automáticamente cuando se precarga la reserva)
idLocal: 4
idPersona: 1

15. estado: finalizada
cupos: 5
calificacion: 4
idReserva: 15 (se genera automáticamente cuando se precarga la reserva)
idLocal: 7
idPersona: 2

16. estado: finalizada
cupos: 5
calificacion: 4
idReserva: 16 (se genera automáticamente cuando se precarga la reserva)
idLocal: 3
idPersona: 2

17. estado: finalizada
cupos: 5
calificacion: 4
idReserva: 17 (se genera automáticamente cuando se precarga la reserva)
idLocal: 1
idPersona: 2

18. estado: finalizada
cupos: 3
calificacion: 4
idReserva: 18 (se genera automáticamente cuando se precarga la reserva)
idLocal: 2
idPersona: 2

19. estado: finalizada

cupos: 4

calificacion: 2

idReserva: 19 (se genera automáticamente cuando se precarga la reserva)

idLocal: 5

idPersona: 1

5. Código

1.3.HTML

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <link rel="icon" href="data:,">
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  </head>
  <body style="background-color: rgb(130, 183, 208)">

    <div id="divBienvenida" style="border-style: solid; border-
width:3px; border-color: blue; font-size: 24px; color:rgb(254, 252, 252);
background-color: rgb(76, 126, 151);">
      <h1 align="center"> BIENVENIDOS AL SISTEMA<br> DE RESERVAS </h1>
    </div>

    <div id="divSctUsuario">
      <h2>Seleccione tipo de usuario al que desea ingresar:</h2>
      <br>
      <input type="button" id="btnIngresoPersona" value="PERSONA"
style="border-style: solid; border-width:4px; border-color: rgb(60, 60,
111); font-size: 40px; color:rgb(247, 247, 247); background-color:
rgb(61, 107, 131);"></td>
      <br><br>
      <input type="button" id="btnIngresoLocal" value="LOCAL"
style="border-style: solid; border-width:4px; border-color: rgb(60, 60,
111); font-size: 40px; color:rgb(249, 244, 244); background-color:
rgb(61, 107, 131);"></td>
    </div>

    <div id="divVolverInicio">
      <input type="button" id="btnVolverInicio" value="Volver al
inicio">
    </div>

    <div id="divVolverPersona">
      <input type="button" id="btnVolverPersona" value="Volver">
    </div>
```

```

<div id="divVolverLocal">
  <input type="button" id="btnVolverLocal" value="Volver">
</div>

<div id="divCerrarSesion">
<input type="button" id="btnCerrarSesion" value="Cerrar Sesión" >
</div>

<div id="divIniciarOregistrarsePersona">
<br><br>
<h1>USTED INGRESÓ A USUARIO PERSONA </h1>
<h2>Seleccione una opción:</h2>
<input type="button" value="REGISTRARSE"
id="btnRegistrarUsuarioPersona">
  <input type="button" value="INGRESAR"
id="btnIniciarSesionUsuarioPersona">
</div>

<div id="divRegistrarseUsuarioPersona">
<br><br>
<h1> USUARIO PERSONA </h1> <br>
<label> COMPLETE LOS SIGUIENTES DATOS PARA REGISTRARSE:
</label><br>
<br><br>
<label for="nombreRegistroUsuarioPersona"> Nombre: </label><input
type="text" id="nombreRegistroUsuarioPersona">
<br>
<label for="usuarioRegistroUsuarioPersona"> Usuario:
</label><input type="text" id="usuarioRegistroUsuarioPersona">
<br>
<label for="contraseñaUsuarioPersona"> Contraseña: </label><input
type="text" id="contraseñaRegistroUsuarioPersona">
<br><br>
<input type="button" value="Registrarse"
id="btnGuardarNuevoUsuarioPersona">
  <p id="pMensajeRegistroUsuarioPersona"></p>
</div>

<div id="divIniciarSesionUsuarioPersona">
<br><br>
<h1> USUARIO PERSONA </h1> <br>
<label> COMPLETE LOS SIGUIENTES DATOS PARA INGRESAR:</label>
<br><br>
<label for="usuarioUsuarioPersona"> Usuario: </label><input
type="text" id="usuarioUsuarioPersona">
<br>

```

```

        <label for="contraseñaUsuarioPersona"> Contraseña: </label><input
type="text" id="contraseñaUsuarioPersona">
        <br><br>
        <input type="button" value="INGRESAR"
id="btnIngresoUsuarioPersona">
        <p id="pMensajeNuevoUsuarioPersona"></p>
    </div>

    <div id="divIniciarSesionUsuarioLocal">
        <br><br>
        <h1> USUARIO LOCAL: </h1> <br>
        <label> COMPLETE LOS SIGUIENTES DATOS PARA INGRESAR </label>
        <br><br>
        <label for="usuarioUsuarioLocal">Usuario: </label><input
type="text" id="usuarioUsuarioLocal">
        <br>
        <label for="contraseñaUsuarioLocal"> Contraseña: </label><input
type="text" id="contraseñaUsuarioLocal">
        <br>
        <br>
        <input type="button" value="INGRESAR"
id="btnIngresoUsuarioLocal">
        <p id="pMensajeInicioUsuarioLocal"></p>
    </div>

    <div id="divYaInicioSesionUsuarioPersona">
        <br><br>
        <label> BIENVENIDO A USUARIO PERSONA </label>
        <br><br>
        <label>Que desea hacer?</label>
        <br><br>
        <input type="button" value="Realizar reserva"
id="btnRealizarReserva">
        <br><br>
        <input type="button" value="Cancelar reserva"
id="btnCancelarReserva">
        <br><br>
        <input type="button" value="Calificar reserva"
id="btnCalificarReserva">
        <br><br>
        <input type="button" value="Ver mis reservas pendientes"
id="btnVerMisReservasPendientes">
        <br><br>
        <input type="button" value="Visualizar informacion estadistica"
id="btnInformacionEstadisticaPersona">
    </div>

```



```

<div id="divYaInicioSesionUsuarioLocal">
  <br><br>
  <label> BIENVENIDO A USUARIO LOCAL </label>
  <br><br>
  <label>Que desea hacer?</label>
  <br><br>
  <input type="button" value="Habilitar/Deshabilitar reservas"
id="btnHabilitarDisponibilidad">
  <br><br>
  <input type="button" value="Cambiar las reserva de estado"
id="btnCambiarLasReservaDeEstado">
  <br><br>
  <input type="button" value="Modificar cupo maximo"
id="btnModificarCupoMaximo">
  <br><br>
  <input type="button" value="Visualizar informacion estadistica"
id="btnInformacionEstadisticaLocal">
</div>

<div id="divHabilitarDeshabilitarReservasUsuarioLocal">
  <br>
  <table border="1">
    <thead>
      <tr>
        <th>Usuario</th>
        <th>Local</th>
        <th>Accion</th>
      </tr>
    </thead>
    <tbody id="bodyTablaHabilitarDeshabilitarReservasUsuarioLoca
l">
      </tbody>
    </table>
  </div>

  <div id="divModificarCuposUsuarioLocal">
    <br>
    <label for="cupoMaximo"> Cupo máximo: <input type="text"
id="cupoMaximo">
    <br><br>
    <input type="button" value="Modificar cupo"
id="btnModificarCupoMaximoDentroDeModificarCupoMaximo">
    <p id="pMensajebtnModificarCuposUsuarioLocal"></p>
  </div>

  <div id="divRealizarReserva">
    <br><br>

```

```

<table border="1">
  <thead>
    <tr>
      <th>Foto</th>
      <th>Local</th>
      <th>Accion</th>
    </tr>
  </thead>
  <tbody id="bodyTablaReservarLocal">
  </tbody>
</table>
</div>

<div id="divCantidadDeCuposEnRealizarReserva">
<br><br>
<label for="cantidadDeCupos"> Cantidad de Cupos: <input
type="number" id="cantidadDeCupos">
<br><br>
<input type="button" value="Confirmar Reserva"
id="btnRealizarReservaDentroDeRealizarReserva">
<p id="pMensajeCantidadDeCuposEnRealizarReserva"></p>
</div>

<div id="divCancelarReserva">
<br><br>
<table border="1">
  <thead>
    <tr>
      <th>Foto</th>
      <th>Local</th>
      <th>Cupos</th>
      <th>Accion</th>
    </tr>
  </thead>
  <tbody id="bodyTablaCancelarReserva">
  </tbody>
</table>
</div>
<div id="divCalificarReservasUsuarioPersona">
<br>
<table border="1">
  <thead>
    <tr>
      <th>Local</th>
      <th>Cupos</th>
      <th>Calificar</th>
    </tr>
  </thead>
  <tbody>
  </tbody>
</table>

```

```

        </thead>
        <tbody id="bodyTablaCalificarReservasUsuarioPersona">
        </tbody>
    </table>
    <p id="pCalificarReserva"></p>
</div>

<div id="divCambiarReservaDeEstadoLocal">
    <br><br>

    <input type="text" id="buscarPersona">
    <input type="button" value="Buscar nombre de usuario"
id="btnBuscarNombreSubcadena" >
    <br><br>
    <input type="button" value="Reestablecer tabla"
id="btnRestablecerTabla" >
    <p id="buscarReservaPorNombre"></p>
    <br>
    <table border="1">
        <thead>
            <tr>
                <th>Persona</th>
                <th>Cupos</th>
                <th>Accion</th>
            </tr>
        </thead>
        <tbody id="bodyCambiarReservaDeEstadoLocal">
        </tbody>
    </table>
</div>

<div id="divVerReservaPendientesUsuarioPersona">
    <br><br>
    <table border="1">
        <thead>
            <tr>
                <th>Foto</th>
                <th>Local</th>
                <th>Cupos</th>
            </tr>
        </thead>
        <tbody id="bodyVerReservaPendientresUsuarioPersona">
        </tbody>
    </table>
</div>

<div id="divInformacionEstadisticaLocal">

```

```

        <br>
        <input type="button" value="Ver el porcentaje de ocupacion del
local" id="btnverElPorcentajeDeOcupacionDelLocal">
        <br><br>
        <input type="button" value="Ver el promedio de calificaciones
del local" id="btnverElPromedioDeCalificacionesDelLocal">
        <br><br>
        <input type="button" value="Ver listado de todos los locales y
el promedio de calificaciones de cada uno"
id="btnverListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUno">
        <br><br>
        <input type="button" value="Ver el total de reservas realizadas
hasta el momento" id="btnVerElTotalDeReservasRealizadasHastaElMomento">
    </div>

    <div id="divverElPorcentajeDeOcupacionDelLocal">
        <br>
        <p id="pverElPorcentajeDeOcupacionDelLocal">
    </div>

    <div id="divverElPromedioDeCalificacionesDelLocal">
        <br>
        <p id="pverElPromedioDeCalificacionesDelLocal">
    </div>

    <div
id="divverListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUno">
        <br>
        <table border="1">
            <thead>
                <tr>
                    <th>Foto</th>
                    <th>Local</th>
                    <th>Promedio de Calificaciones</th>
                </tr>
            </thead>
            <tbody id="bodyverListadoDeTodosLosLocalesYElPromedioDeCalific
acionesDeCadaUno">
            </tbody>
        </table>
    </div>

    <div id="divVerElTotalDeReservasRealizadasHastaElMomento">
        <br>
        <p id="pVerElTotalDeReservasRealizadasHastaElMomento">
    </div>

```

```

<div id="divInformacionEstadisticaPersona">
  <br>
  <input type="button" value="Ver el porcentaje de reservas"
id="btnverElPorcentajeDeReservas">
  <br><br>
  <input type="button" value="Ver local donde hice mas reservas"
id="btnverElLocalDondeHaHechoMasReservas">
</div>

<div id="divListadoReservasUsuarioPersona">
  <br>
  <table border="1">
    <thead>
      <tr>
        <th>Local con reservas completadas</th>
        <th>Mis reservas</th>
        <th>Cantidad de reservas que tiene el local</th>
        <th>Porcentaje de mis reservas en este local</th>
      </tr>
    </thead>
    <tbody id="bodyListadoReservasUsuarioPersona">
    </tbody>
  </table>
</div>

<div id="divverElLocalDondeHaHechoMasReservas">
  <br>
  <table border="1">
    <thead>
      <tr>
        <th>Local</th>
        <th>Reservas</th>
      </tr>
    </thead>
    <tbody id="bodyverElLocalDondeHaHechoMasReservas">
    </tbody>
  </table>
</div>

<script type="text/javascript" src="js/clases.js"></script>

```

```
<script type="text/javascript" src="js/miCodigo.js"></script>  
</body>  
</html>
```

1.4.JAVASCRIPT

```
let proximoIdTipoUsuarioPersona = 1;
let proximoIdTipoUsuarioLocal = 1;
let proximaReserva = 1;

class UsuariosPersona {
  constructor (pNombre, pNombreUsuario, pPassword) {
    this.nombre = pNombre;
    this.nombreUsuario = pNombreUsuario;
    this.password = pPassword;
    this.idPersona = proximoIdTipoUsuarioPersona;

    proximoIdTipoUsuarioPersona++;
  }
}

class UsuariosLocal {
  constructor (pNombre, pNombreUsuario, pPassword, pTipoDeLocal,
pDireccion, pCupoMaximo, pFoto, pHabilitado) {
    this.nombre = pNombre;
    this.nombreUsuario = pNombreUsuario;
    this.password = pPassword;
    this.tipoDeLocal = pTipoDeLocal;
    this.direccion = pDireccion;
    this.cupoMaximo = pCupoMaximo;
    this.foto = pFoto;
    this.habilitado = pHabilitado;
    this.idLocal = proximoIdTipoUsuarioLocal;

    proximoIdTipoUsuarioLocal++;
  }
}

class Reservas {
  constructor (pEstado, pCupos, pCalificacion, pidLocal, pidPersona) {
    this.estado = pEstado;
    this.cupos = pCupos;
    this.calificacion = pCalificacion;
    this.idReserva = proximaReserva;
    this.idLocal = pidLocal;
    this.idPersona = pidPersona;

    proximaReserva++;
  }
}
```

```

let usuariosPersona = [];
let usuariosLocal = [];
let reservas = [];
let nombreUsuarioLogueado = null;
let nombreUsuarioLocalSeleccionadoDentroDeRealizarReserva = null;

inicializar();

function inicializar() {
    mostrarPantallaPrincipal();
    cargarClickEnBotones();
    precargarDatosPersonas();
    precargarDatosLocales();
    precargarDatosReservas();
}

function mostrarPantallaPrincipal() {
    ocultarPantallas();
    document.querySelector("#divBienvenida").style.display = "block";
    document.querySelector("#divSctUsuario").style.display = "block";
}

function cargarClickEnBotones() {
    document.querySelector("#btnCerrarSesion").addEventListener("click",
    btnCerrarSesionHandler);
    document.querySelector("#btnVolverInicio").addEventListener("click",
    btnVolverInicioHandler);
    document.querySelector("#btnVolverPersona").addEventListener("click",
    btnVolverPersonaHandler);
    document.querySelector("#btnVolverLocal").addEventListener("click",
    btnVolverLocalHandler);
    document.querySelector("#btnIngresoPersona").addEventListener("click",
    btnIngresoPersonaHandler);
    document.querySelector("#btnIngresoLocal").addEventListener("click",
    btnIngresoLocalHandler);
    document.querySelector("#btnRegistrarUsuarioPersona").addEventListener("click",
    btnRegistrarUsuarioPersonaHandler);

    document.querySelector("#btnIniciarSesionUsuarioPersona").addEventListener("click",
    btnIniciarSesionUsuarioPersonaHandler);

    document.querySelector("#btnGuardarNuevoUsuarioPersona").addEventListener("click",
    btnGuardarNuevoUsuarioPersonaHandler);
    document.querySelector("#btnIngresoUsuarioPersona").addEventListener("click",
    btnIngresoUsuarioPersonaHandler);
    document.querySelector("#btnIngresoUsuarioLocal").addEventListener("click",
    btnIngresoUsuarioLocalHandler);

```



```

    document.querySelector("#btnHabilitarDisponibilidad").addEventListener("click",
    btnHabilitarDisponibilidadHandler);
    document.querySelector("#btnModificarCupoMaximo").addEventListener("click",
    btnModificarCupoMaximoHandler);

    document.querySelector("#btnModificarCupoMaximoDentroDeModificarCupoMaximo
    ").addEventListener("click",
    btnModificarCupoMaximoDentroDeModificarCupoMaximoHandler);
    document.querySelector("#btnRealizarReserva").addEventListener("click",
    btnrealizarReservaHandler);

    document.querySelector("#btnRealizarReservaDentroDeRealizarReserva").addEventLis
    tener("click", btnRealizarReservaDentroDeRealizarReservaHandler);
    document.querySelector("#btnCancelarReserva").addEventListener("click",
    btnCancelarReservaHandler);

    document.querySelector("#btnCambiarLasReservaDeEstado").addEventListener("click"
    , btnCambiarLasReservaDeEstadoHandler);
    document.querySelector("#btnCalificarReserva").addEventListener("click",
    btnCalificarReservaHandler);

    document.querySelector("#btnVerMisReservasPendientes").addEventListener("click",
    btnVerMisReservasPendientesHandler);
    document.querySelector("#btnBuscarNombreSubcadena").addEventListener("click",
    btnBuscarNombreSubcadenaHandler);
    document.querySelector("#btnRestablecerTabla").addEventListener("click",
    btnRestablecerTablaHandler);

    document.querySelector("#btnInformacionEstadisticaLocal").addEventListener("click",
    btnInformacionEstadisticaLocalHandler);

    document.querySelector("#btnverElPorcentajeDeOcupacionDelLocal").addEventListen
    er("click", btnverElPorcentajeDeOcupacionDelLocalHandler);

    document.querySelector("#btnverElPromedioDeCalificacionesDelLocal").addEventList
    ener("click", btnverElPromedioDeCalificacionesDelLocalHandler);

    document.querySelector("#btnverListadoDeTodosLosLocalesYElPromedioDeCalificaci
    onesDeCadaUno").addEventListener("click",
    btnverListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUnoHandler);

    document.querySelector("#btnVerElTotalDeReservasRealizadasHastaElMomento").add
    EventListener("click", btnVerElTotalDeReservasRealizadasHastaElMomentoHandler);

    document.querySelector("#btnInformacionEstadisticaPersona").addEventListener("clie
    k", btnInformacionEstadisticaPersonaHandler);

```

```

document.querySelector("#btnverElPorcentajeDeReservas").addEventListener("click",
btnverElPorcentajeDeReservasHandler);

document.querySelector("#btnverElLocalDondeHaHechoMasReservas").addEventListener(
ner("click", btnverElLocalDondeHaHechoMasReservasHandler);
}

function btnCerrarSesionHandler () {
    ocultarPantallas();
    vaciarCampos();
    mostrarPantallaPrincipal();
    nombreUsuarioLogueado = null;
}

function btnVolverInicioHandler() {
    mostrarPantallaPrincipal();
}

function btnVolverPersonaHandler() {
    mostrarYaInicioSesionUsuarioPersona();
}

function btnVolverLocalHandler() {
    mostrarYaInicioSesionUsuarioLocal();
}

function btnIngresoPersonaHandler() {
    mostrarIngresoPersona();
    document.querySelector("#btnVolverInicio").style.display = "block";
}

function btnIngresoLocalHandler() {
    mostrarIngresoLocal();
    document.querySelector("#btnVolverInicio").style.display = "block";
}

function btnRegistrarUsuarioPersonaHandler() {
    mostrarRegistroPersona();
}

function ocultarPantallas() {
    vaciarCampos();
    borrarMensajes();
    document.querySelector("#divBienvenida").style.display = "none";
    document.querySelector("#divSctUsuario").style.display = "none";
}

```

```

document.querySelector("#divIniciarOregistrarsePersona").style.display = "none";
document.querySelector("#divRegistrarseUsuarioPersona").style.display = "none";
document.querySelector("#divIniciarSesionUsuarioPersona").style.display = "none";
document.querySelector("#divCerrarSesion").style.display = "none";
document.querySelector("#divIniciarSesionUsuarioLocal").style.display = "none";
document.querySelector("#divVolverInicio").style.display = "none";
document.querySelector("#divVolverPersona").style.display = "none";
document.querySelector("#divVolverLocal").style.display = "none";
document.querySelector("#divYaInicioSesionUsuarioPersona").style.display =
"none";
document.querySelector("#divYaInicioSesionUsuarioLocal").style.display = "none";

document.querySelector("#divHabilitarDeshabilitarReservasUsuarioLocal").style.display = "none";
document.querySelector("#divRealizarReserva").style.display = "none";
document.querySelector("#divModificarCuposUsuarioLocal").style.display = "none";
document.querySelector("#divCantidadDeCuposEnRealizarReserva").style.display =
"none";
document.querySelector("#divCancelarReserva").style.display = "none";
document.querySelector("#divCalificarReservasUsuarioPersona").style.display =
"none";
document.querySelector("#divCambiarReservaDeEstadoLocal").style.display =
"none";
document.querySelector("#divVerReservaPendientesUsuarioPersona").style.display =
"none";

document.querySelector("#divHabilitarDeshabilitarReservasUsuarioLocal").style.display = "none";
document.querySelector("#divInformacionEstadisticaLocal").style.display = "none";
document.querySelector("#divverElPorcentajeDeOcupacionDelLocal").style.display =
"none";

document.querySelector("#divverElPromedioDeCalificacionesDelLocal").style.display =
"none";

document.querySelector("#divverListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUno").style.display = "none";

document.querySelector("#divVerElTotalDeReservasRealizadasHastaElMomento").style.display = "none";
document.querySelector("#divInformacionEstadisticaPersona").style.display =
"none";
document.querySelector("#divListadoReservasUsuarioPersona").style.display =
"none";

document.querySelector("#divverElLocalDondeHaHechoMasReservas").style.display =
"none";
}

```

```

function vaciarCampos() {
    document.querySelector("#nombreRegistroUsuarioPersona").value = "";
    document.querySelector("#usuarioRegistroUsuarioPersona").value = "";
    document.querySelector("#contraseñaRegistroUsuarioPersona").value = "";
    document.querySelector("#usuarioUsuarioPersona").value = "";
    document.querySelector("#contraseñaUsuarioPersona").value = "";
    document.querySelector("#usuarioUsuarioLocal").value = "";
    document.querySelector("#contraseñaUsuarioLocal").value = "";
    document.querySelector("#cantidadDeCupos").value = "";
    document.querySelector("#nombreRegistroUsuarioPersona").value = "";
    document.querySelector("#cupoMaximo").value = "";
    document.querySelector("#buscarPersona").value = "";
}

function borrarMensajes() {
    document.querySelector("#pMensajeRegistroUsuarioPersona").innerHTML = "";
    document.querySelector("#pMensajeNuevoUsuarioPersona").innerHTML = "";
    document.querySelector("#pMensajeInicioUsuarioLocal").innerHTML = "";

    document.querySelector("#pMensajeCantidadDeCuposEnRealizarReserva").innerHTML = "";
    document.querySelector("#buscarReservaPorNombre").innerHTML = "";
    document.querySelector("#pMensajebtnModificarCuposUsuarioLocal").innerHTML = "";
    document.querySelector("#pCalificarReserva").innerHTML = "";
}

function mostrarIngresoPersona() {
    ocultarPantallas();
    document.querySelector("#divIniciarOregistrarsePersona").style.display = "block";
    document.querySelector("#divVolverInicio").style.display = "block";
}

function mostrarIngresoLocal() {
    ocultarPantallas();
    document.querySelector("#divIniciarSesionUsuarioLocal").style.display = "block";
    document.querySelector("#divVolverInicio").style.display = "block";
}

function mostrarRegistroPersona() {
    ocultarPantallas();
    document.querySelector("#divRegistrarseUsuarioPersona").style.display = "block";
    document.querySelector("#divVolverInicio").style.display = "block";
}

function mostrarInicioSesionPersona() {
    ocultarPantallas();

```

```

        document.querySelector("#divIniciarSesionUsuarioPersona").style.display = "block";
        document.querySelector("#divVolverInicio").style.display = "block";
    }

    function mostrarCerrarSesion(){
        document.querySelector("#divCerrarSesion").style.display = "block";
    }

    function mostrarYaInicioSesionUsuarioPersona() {
        ocultarPantallas();
        mostrarCerrarSesion();
        document.querySelector("#divYaInicioSesionUsuarioPersona").style.display =
"block";
        document.querySelector("#btnVolverInicio").style.display = "none";
    }

    function mostrarYaInicioSesionUsuarioLocal() {
        ocultarPantallas();
        mostrarCerrarSesion();
        document.querySelector("#divYaInicioSesionUsuarioLocal").style.display = "block";
        document.querySelector("#btnVolverInicio").style.display = "none";
    }

    function mostrarHabilitarDeshabilitarReservas() {
        ocultarPantallas();
        mostrarCerrarSesion();
        mostrarVolverLocal();

        document.querySelector("#divHabilitarDeshabilitarReservasUsuarioLocal").style.display =
"block";
        document.querySelector("#btnVolverInicio").style.display = "none";
    }

    function mostrarModificarCupo() {
        ocultarPantallas();
        mostrarCerrarSesion();
        mostrarVolverLocal();
        document.querySelector("#divModificarCuposUsuarioLocal").style.display =
"block";
        document.querySelector("#btnVolverInicio").style.display = "none";
    }

    function mostrarRealizarReserva() {
        ocultarPantallas();
        mostrarCerrarSesion();
        mostrarVolverPersona();
        document.querySelector("#divRealizarReserva").style.display = "block";
    }

```

```

    document.querySelector("#btnVolverInicio").style.display = "none";
}

function mostrarVolverPersona() {
    document.querySelector("#divVolverPersona").style.display = "block";
}

function mostrarVolverLocal() {
    document.querySelector("#divVolverLocal").style.display = "block";
}

function mostrarCantidadDeCuposEnRealizarReserva() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
    vaciarCampos();
    document.querySelector("#divCantidadDeCuposEnRealizarReserva").style.display =
"block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

function mostrarCancelarReserva() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
    document.querySelector("#divCancelarReserva").style.display = "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

function mostrarCambiarReservaDeEstadoLocal() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();
    vaciarCampos();
    borrarMensajes();
    document.querySelector("#divCambiarReservaDeEstadoLocal").style.display =
"block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//FUNCION QUE PRECARGA DATOS DE PERSONAS
function precargarDatosPersonas() {
    if (validarDatosPrecargaPersonas("Ana","UP1","An1234") == true) {
        guardarNuevoUsuarioPersonaArray("Ana","UP1","An1234");
    }
    if (validarDatosPrecargaPersonas("Emiliano","UP2","Em1234") == true) {
        guardarNuevoUsuarioPersonaArray("Emiliano","UP2","Em1234");
    }
}

```

```

    if (validarDatosPrecargaPersonas("Juan","UP3","Ju1234") == true) {
        guardarNuevoUsuarioPersonaArray("Juan","UP3","Ju1234");
    }
    if (validarDatosPrecargaPersonas("Andres","UP4","An1234") == true) {
        guardarNuevoUsuarioPersonaArray("Andres","UP4","An1234");
    }
    if (validarDatosPrecargaPersonas("Florencia","UP5","Fl1234") == true) {
        guardarNuevoUsuarioPersonaArray("Florencia","UP5","Fl1234");
    }
    if (validarDatosPrecargaPersonas("Bruno","UP6","Br1234") == true) {
        guardarNuevoUsuarioPersonaArray("Bruno","UP6","Br1234");
    }
    if (validarDatosPrecargaPersonas("Luis","UP7","Lu1234") == true) {
        guardarNuevoUsuarioPersonaArray("Luis","UP7","Lu1234");
    }
}

//FUNCION QUE VALIDA PRECARGA DE PERSONAS
function validarDatosPrecargaPersonas(nombre, usuario, password) {
    if (nombre == "" || !isNaN(nombre) || usuario == "" || password == "" ||
    buscarNombreUsuario(usuarioPersona, usuario) != null ||
    buscarNombreUsuario(usuarioLocal, usuario) != null || validarPassword(password) ==
    false) {
        resultado = false;
    }
    else {
        resultado = true;
    }
    return resultado;
}

//FUNCION QUE PRECARGA DATOS DE LOCALES
function precargarDatosLocales() {
    if (validarDatosPrecargaLocales("Teatro Solis","UL1", "Ts1234", "Teatro",
    "Reconquista S/N", 1500, "teatroSolis.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("Teatro Solis","UL1", "Ts1234", "Teatro",
    "Reconquista S/N", 1500, "teatroSolis.jpg", true);
    }
    if (validarDatosPrecargaLocales("El Galpon","UL2","Eg1234", "Teatro", "Av. 18 de
    julio 1618", 800, "El-galpon.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("El Galpon","UL2","Eg1234", "Teatro", "Av. 18
    de julio 1618", 800, "El-galpon.jpg", true);
    }
    if (validarDatosPrecargaLocales("Museo Juan Manuel Blanes","UL3","Mj1234",
    "Museo", "Millán 4015", 50, "Sala_blanes.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("Museo Juan Manuel Blanes","UL3","Mj1234",
    "Museo", "Millán 4015", 50, "Sala_blanes.jpg", true);
    }
}

```

```

    if (validarDatosPrecargaLocales("Museo Artes Visuales","UL4", "Mv1234",
    "Museo", "Av. Tomas Giribaldi 2283", 100, "museo_artes_visuales.jpg", true) == true)
    {
        guardarNuevoUsuarioLocalArray("Museo Artes Visuales","UL4", "Mv1234",
    "Museo", "Av. Tomas Giribaldi 2283", 100, "museo_artes_visuales.jpg", true);
    }
    if (validarDatosPrecargaLocales("Plantado","UL5","Rp1234", "Restaurante", "Rbla.
    República del Perú 1479", 100, "restaurante-plantado.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("Plantado","UL5","Rp1234", "Restaurante",
    "Rbla. República del Perú 1479", 100, "restaurante-plantado.jpg", true);
    }
    if (validarDatosPrecargaLocales("Garcia","UL6","Rg1234", "Restaurante", "Av.
    Alfredo Arocena 1587", 100, "garcia.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("Garcia","UL6","Rg1234", "Restaurante", "Av.
    Alfredo Arocena 1587", 100, "garcia.jpg", true);
    }
    if (validarDatosPrecargaLocales("Café Misterio","UL7","Cm1234", "Restaurante",
    "Costa Rica 1700", 50, "cafe_misterio.jpg", true) == true) {
        guardarNuevoUsuarioLocalArray("Café Misterio","UL7","Cm1234",
    "Restaurante", "Costa Rica 1700", 50, "cafe_misterio.jpg", true);
    }
}

```

//FUNCION QUE VALIDA PRECARGA DE LOCALES

```

function validarDatosPrecargaLocales(nombre, usuario, password, tipoDeLocal,
direccion, cupoMaximo, foto, habilitado) {
    if (nombre == "" || !isNaN(nombre) || usuario == "" || password == "" || tipoDeLocal
    == "" || !isNaN(tipoDeLocal) || direccion == "" || !isNaN(direccion) || cupoMaximo ==
    "" || cupoMaximo < 0 || cupoMaximo == 0 || isNaN(cupoMaximo) || foto == "" ||
    !isNaN(foto) || habilitado == "" || (habilitado != true && habilitado != false) ||
    buscarNombreUsuario(usuariosLocal, usuario) != null ||
    buscarNombreUsuario(usuariosPersona, usuario) != null) {
        resultado = false;
    }
    else {
        resultado = true;
    }
    return resultado;
}

```

//FUNCION QUE PRECARGA DATOS DE RESERVAS

```

function precargarDatosReservas() {
    if (validarDatosPrecargaReservas("pendiente",5,null,1,1) == true &&
    buscarNombreUsuario(usuariosLocal, "UL1") != null &&
    buscarNombreUsuario(usuariosPersona, "UP1") != null &&
    cuentaReservasPendientesPorUsuarioPersona("UP1", "UL1") == 0) {
        guardarNuevoUsuarioReservasArray("pendiente",5,null,1,1);
    }
}

```



```

        if (validarDatosPrecargaReservas("pendiente",4,null,2,1) == true &&
        buscarNombreUsuario(usuariosLocal, "UL2") != null &&
        buscarNombreUsuario(usuariosPersona, "UP1") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP1", "UL2") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",4,null,2,1);
        }

        if (validarDatosPrecargaReservas("pendiente",6,null,2,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL2") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP2", "UL2") == 0){
            guardarNuevoUsuarioReservasArray("pendiente",6,null,2,2);
        }

        if (validarDatosPrecargaReservas("pendiente",5,null,4,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL4") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP2", "UL4") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",5,null,4,2);
        }

        if (validarDatosPrecargaReservas("pendiente",10,null,3,3) == true &&
        buscarNombreUsuario(usuariosLocal, "UL3") != null &&
        buscarNombreUsuario(usuariosPersona, "UP3") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP3", "UL3") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",10,null,3,3);
        }

        if (validarDatosPrecargaReservas("pendiente",5,null,5,3) == true &&
        buscarNombreUsuario(usuariosLocal, "UL5") != null &&
        buscarNombreUsuario(usuariosPersona, "UP3") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP3", "UL5") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",5,null,5,3);
        }

        if (validarDatosPrecargaReservas("pendiente",2,null,4,4) == true &&
        buscarNombreUsuario(usuariosLocal, "UL4") != null &&
        buscarNombreUsuario(usuariosPersona, "UP4") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP4", "UL4") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",2,null,4,4);
        }

        if (validarDatosPrecargaReservas("pendiente",3,null,7,4) == true &&
        buscarNombreUsuario(usuariosLocal, "UL7") != null &&
        buscarNombreUsuario(usuariosPersona, "UP4") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP4", "UL7") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",3,null,7,4);
        }

        if (validarDatosPrecargaReservas("pendiente",5,null,6,4) == true &&
        buscarNombreUsuario(usuariosLocal, "UL6") != null &&
        buscarNombreUsuario(usuariosPersona, "UP4") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP4", "UL6") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",5,null,6,4);
        }

```

```

        if (validarDatosPrecargaReservas("pendiente",7,null,5,5) == true &&
        buscarNombreUsuario(usuariosLocal, "UL5") != null &&
        buscarNombreUsuario(usuariosPersona, "UP5") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP5", "UL5") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",7,null,5,5);
        }

        if (validarDatosPrecargaReservas("pendiente",5,null,2,5) == true &&
        buscarNombreUsuario(usuariosLocal, "UL2") != null &&
        buscarNombreUsuario(usuariosPersona, "UP5") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP5", "UL2") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",5,null,2,5);
        }

        if (validarDatosPrecargaReservas("pendiente",5,null,1,5) == true &&
        buscarNombreUsuario(usuariosLocal, "UL1") != null &&
        buscarNombreUsuario(usuariosPersona, "UP5") != null &&
        cuentaReservasPendientesPorUsuarioPersona("UP5", "UL1") == 0) {
            guardarNuevoUsuarioReservasArray("pendiente",5,null,1,5);
        }

        if (validarDatosPrecargaReservas("finalizada",4,3,6,1) == true &&
        buscarNombreUsuario(usuariosLocal, "UL6") != null &&
        buscarNombreUsuario(usuariosPersona, "UP1") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",4,3,6,1);
        }

        if (validarDatosPrecargaReservas("finalizada",4,3,4,1) == true &&
        buscarNombreUsuario(usuariosLocal, "UL4") != null &&
        buscarNombreUsuario(usuariosPersona, "UP1") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",4,3,4,1);
        }

        if (validarDatosPrecargaReservas("finalizada",5,4,7,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL7") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",5,4,7,2);
        }

        if (validarDatosPrecargaReservas("finalizada",5,4,3,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL3") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",5,4,3,2);
        }

        if (validarDatosPrecargaReservas("finalizada",5,4,1,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL1") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",5,4,1,2);
        }

        if (validarDatosPrecargaReservas("finalizada",3,4,2,2) == true &&
        buscarNombreUsuario(usuariosLocal, "UL2") != null &&
        buscarNombreUsuario(usuariosPersona, "UP2") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",3,4,2,2);
        }

```

```

        if (validarDatosPrecargaReservas("finalizada",4,2,5,1) == true &&
        buscarNombreUsuario(usuariosLocal, "UL5") != null &&
        buscarNombreUsuario(usuariosPersona, "UP1") != null) {
            guardarNuevoUsuarioReservasArray("finalizada",4,2,5,1);
        }
    }

//FUNCION QUE VALIDA PRECARGA DE RESERVAS
function validarDatosPrecargaReservas(estado, cupos, calificacion, idlocal, idPersona) {
    if (estado == "" || !isNaN(estado) || cupos == "" || cupos == 0 || cupos < 0 ||
    isNaN(cupos) || calificacion < 0 || calificacion == 0 || isNaN(calificacion) || idlocal == ""
    || isNaN(idlocal) || idPersona == "" || isNaN(idPersona) || idlocal < 0 || idPersona < 0
    || idlocal == 0 || idPersona == 0) {
        resultado = false;
    }
    else {
        resultado = true;
    }
    return resultado;
}

//FUNCION QUE GUARDA EL NUEVO USUARIO PERSONA
function btnGuardarNuevoUsuarioPersonaHandler() {
    borrarMensajes();
    let mensaje = "";

    const nombreIngresadoRegistroPersona =
    document.querySelector("#nombreRegistroUsuarioPersona").value;
    const usuarioIngresadoRegistroPersona =
    document.querySelector("#usuarioRegistroUsuarioPersona").value;
    const passwordIngresadoRegistroPersona =
    document.querySelector("#contraseñaRegistroUsuarioPersona").value;

    //ESTAS VARIABLES ME DEVUELVEN SI YA EXISTE USUARIO
    REGISTRADO CON ESE NOMBRE DE USUARIO
    let usuarioLocalEncontrado = buscarNombreUsuario(usuariosLocal,
    usuarioIngresadoRegistroPersona);
    let usuarioPersonaEncontrado = buscarNombreUsuario(usuariosPersona,
    usuarioIngresadoRegistroPersona);

    let validacionPassword=validarPassword(passwordIngresadoRegistroPersona);

    // NULL = NO EXISTE AUN USUARIO CON ESE NOMBRE DE USUARIO

    if (!isNaN(nombreIngresadoRegistroPersona) || nombreIngresadoRegistroPersona ==
    "" || usuarioIngresadoRegistroPersona == "" || passwordIngresadoRegistroPersona ==
    "") {
        mensaje= "Se deben completar todos los campos con informacion valida";
    }
}

```

```

    } else if (usuarioLocalEncontrado == null && usuarioPersonaEncontrado == null
    && validacionPassword== true) {
        // FUNCION ingresarNomYcontraUsuarioPersona PUSHEA EL NUEVO
        USUARIO Y CONTRASEÑA
        guardarNuevoUsuarioPersonaArray(nombreIngresadoRegistroPersona,
        usuarioIngresadoRegistroPersona, passwordIngresadoRegistroPersona);
        mensaje = "Usuario registrado";
        vaciarCampos();
    } else if (usuarioLocalEncontrado == null && usuarioPersonaEncontrado == null
    && validacionPassword== false){
        mensaje= "La contraseña debe contener al menos 6 caracteres, una minúscula, una
        mayúscula y un número.";
    } else {
        mensaje = "El usuario ya existe";
    }
    document.querySelector("#pMensajeRegistroUsuarioPersona").innerHTML =
    mensaje;
}

//FUNCION QUE VALIDA CONTRASEÑA
function validarPassword(passwordIngresado){
    let cumplePassword= false;
    let tieneMayuscula= false;
    let tieneMinuscula= false;
    let tieneNumero= false;
    if (passwordIngresado.length> 5){
        for(let i=0; i<passwordIngresado.length; i++){
            if(passwordIngresado.charCodeAt(i) >=65 &&
            passwordIngresado.charCodeAt(i) <=90){
                tieneMayuscula=true;
            }
            if(passwordIngresado.charCodeAt(i) >=97 &&
            passwordIngresado.charCodeAt(i) <=122){
                tieneMinuscula=true;
            }
            if(passwordIngresado.charCodeAt(i) >=48 &&
            passwordIngresado.charCodeAt(i) <=57){
                tieneNumero=true;
            }
        }
    }
    if (tieneMayuscula && tieneMinuscula && tieneNumero){
        cumplePassword= true;
    }

    return cumplePassword;
}

```

```
// CHEQUEO QUE LOS DATOS PARA REGISTRAR USUARIO NO EXISTA
//buscarNombreUsuarioPersona FUNCION QUE RECIBE ARRAY DE USUARIOS Y
NOMBRE DE USUARIO INGRESADO
```

```
function buscarNombreUsuario(usuarios, nombreUsuarioIngresado){
    let resultado = null;
    let i = 0;
    while (resultado == null && i < usuarios.length) {
        let usuarioActual = usuarios[i];
        if (usuarioActual.nombreUsuario.toLowerCase() ==
nombreUsuarioIngresado.toLowerCase()) {
            resultado = usuarioActual;
        }
        i++;
    }
    return resultado;
}
```

```
//FUNCION QUE BUSCA RESERVA POR NUMERO IDENTIFICADOR
```

```
function buscarReservaPorId(usuarios, Id){
    let resultado = null;
    let i = 0;
    while (resultado == null && i < usuarios.length) {
        let usuarioActual = usuarios[i];
        if (usuarioActual.idReserva == Id) {
            resultado = usuarioActual;
        }
        i++;
    }
    return resultado;
}
```

```
//FUNCION QUE RECIBE NOMBRE, NOMBRE DE USUARIO Y CONTRASEÑA
INGRESADA POR USUARIO PERSONA Y GUARDA
```

```
function guardarNuevoUsuarioPersonaArray(nombrePersona, usuarioPersona,
passwordPersona) {
    let nuevoUsuario = new UsuariosPersona(nombrePersona, usuarioPersona,
passwordPersona);
    usuariosPersona.push(nuevoUsuario);
}
```

```
//FUNCION QUE RECIBE NOMBRE, NOMBRE DE USUARIO, CONTRASEÑA,
TIPO DE LOCAL, DIRECCION, CUPO MAXIMO, FOTO Y ESTADO
```

```
HABILITADO O NO INGRESADA POR USUARIO LOCAL Y GUARDA
function guardarNuevoUsuarioLocalArray(nombreLocal, usuarioLocal, passwordLocal,
tipoDeLocal, direccion, cupoMaximo, foto, habilitado) {
    let nuevoUsuario = new UsuariosLocal(nombreLocal, usuarioLocal, passwordLocal,
tipoDeLocal, direccion, cupoMaximo, foto, habilitado);
    usuariosLocal.push(nuevoUsuario);
}
```

```

}

//FUNCION QUE RECIBE ESTADO (PENDIENTE O FINRALIZADA) Y NOMBRE
USUARIO PERSONA LOGUEADA Y GUARDA
function guardarNuevoUsuarioReservasArray(estado, cupos, calificacion, idLocal,
idPersona) {
    let nuevoUsuario = new Reservas (estado, cupos, calificacion, idLocal, idPersona);
    reservas.push(nuevoUsuario);
}

//LA PERSONA SELECCIONO INICIAR SESION
function btnIniciarSesionUsuarioPersonaHandler() {
    mostrarInciarSesionPersona();
}

//PERSONA INGRESA DATOS E INGRESA
function btnIngresoUsuarioPersonaHandler () {
    borrarMensajes();
    let mensaje = "";

    const usuarioIngresadoPersona =
document.querySelector("#usuarioUsuarioPersona").value;
    const passwordIngresadoPersona =
document.querySelector("#contraseñaUsuarioPersona").value;

    let usuarioPersonaEncontrado = buscarNombreUsuario(usuarioPersona,
usuarioIngresadoPersona);
    let usuarioIngresadoPersonaMinuscula= usuarioIngresadoPersona.toLowerCase();

    if (usuarioIngresadoPersona=="" || passwordIngresadoPersona=="") {
        document.querySelector("#contraseñaUsuarioPersona").value = "";
        mensaje= "Se deben completar todos los campos";
    } else if (usuarioPersonaEncontrado == null ) {
        mensaje = "El usuario no existe";
    } else if (usuarioPersonaEncontrado != null && usuarioIngresadoPersonaMinuscula
== usuarioPersonaEncontrado.nombreUsuario.toLowerCase() &&
usuarioPersonaEncontrado.password == passwordIngresadoPersona) {
        vaciarCampos();
        mostrarYaInicioSesionUsuarioPersona();
        nombreUsuarioLogueado=usuarioIngresadoPersona;
    } else {
        document.querySelector("#contraseñaUsuarioPersona").value = "";
        mensaje= "Contraseña incorrecta, intente nuevamente"
    }
    document.querySelector("#pMensajeNuevoUsuarioPersona").innerHTML = mensaje;
}

```

```

//LOCAL INGRESA DATOS E INGRESA
function btnIngresoUsuarioLocalHandler(){
    borrarMensajes();
    let mensaje = "";

    const usuarioIngresadoLocal =
document.querySelector("#usuarioUsuarioLocal").value;
    const passwordIngresadoLocal =
document.querySelector("#contraseñaUsuarioLocal").value;

    let usuarioLocalEncontrado = buscarNombreUsuario(usuariosLocal,
usuarioIngresadoLocal);

    if (usuarioIngresadoLocal=="" || passwordIngresadoLocal=="") {
        document.querySelector("#contraseñaUsuarioLocal").value = "";
        mensaje= "Se deben completar todos los campos";
    } else if (usuarioLocalEncontrado == null ) {
        mensaje = "El usuario no existe";
    } else if (usuarioLocalEncontrado != null &&
usuarioLocalEncontrado.nombreUsuario.toLowerCase() ==
usuarioIngresadoLocal.toLowerCase() && usuarioLocalEncontrado.password ==
passwordIngresadoLocal){
        vaciarCampos();
        mostrarYaInicioSesionUsuarioLocal();
        nombreUsuarioLogueado=usuarioIngresadoLocal;
    } else {
        document.querySelector("#contraseñaUsuarioLocal").value = "";
        mensaje= "Contraseña incorrecta, intente nuevamente"
    }
    document.querySelector("#pMensajeInicioUsuarioLocal").innerHTML = mensaje;
}

function btnHabilitarDisponibilidadHandler() {
    mostrarHabilitarDeshabilitarReservas();
    TablaHabilitarDeshabilitar();
}

//HABILITAR/DESHABILITAR EL LOCAL LOGUEADO. GENERO UNA TABLA
A PARTIR DE LOS DATOS DEL USUARIO LOGUEADO
function TablaHabilitarDeshabilitar() {
    let bodyHTML = "";
    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);
    let textoBoton = "Deshabilitar";
    if (!usuarioLocallogueado.habilitado) { // usuarioLocallogueado.habilitado == false;
        textoBoton = "Habilitar";
    }
    bodyHTML += `

```

```

        <tr>
            <td>
                ${usuarioLocallogueado.nombreUsuario}
            </td>
            <td>
                ${usuarioLocallogueado.nombre}
            </td>
            <td>
                <input data-usuario="${usuarioLocallogueado.nombreUsuario}"
type="button" value="${textoBoton}" id="btnmodificarDisponibilidadLocal">
            </td>
        </tr>
    `;

document.querySelector("#bodyTablaHabilitarDeshabilitarReservasUsuarioLocal").inn
erHTML = bodyHTML;

document.querySelector("#btnmodificarDisponibilidadLocal").addEventListener("click
", btnmodificarDisponibilidadLocalHandler);
}

//APLICO LA FUNCION DE MODIFICAR LA DISPONIBILIDAD DEL LOCAL
function btnmodificarDisponibilidadLocalHandler() {
    let nombreUsuarioParaCambiarEstado = this.getAttribute("data-usuario");
    modificarDisponibilidadLocal(nombreUsuarioParaCambiarEstado);
    TablaHabilitarDeshabilitar();
}

//MODIFICO LA DISPONIBILIDAD DEL LOCAL
function modificarDisponibilidadLocal(nombreUsuarioLogueado) {
    let usuarioLocallogueado = buscarNombreUsuario(usuarioLocal,
nombreUsuarioLogueado);
    usuarioLocallogueado.habilitado = !usuarioLocallogueado.habilitado;
}

//FUNCION QUE LLAMA A FUNCION QUE MODIFICA CUPO MAXIMO
function btnModificarCupoMaximoHandler() {
    mostrarModificarCupo();
    vaciarCampos();
    borrarMensajes();
}

//MODIFICO CUPO MAXIMO
function btnModificarCupoMaximoDentroDeModificarCupoMaximoHandler () {
    let cupoIngresado = parseInt(document.querySelector("#cupoMaximo").value);
    let usuarioLocallogueado = buscarNombreUsuario(usuarioLocal,
nombreUsuarioLogueado);

```



```

    if (localTieneReservasPendientesEnEstadoPendiente(usuarioLocallogueado)==0 &&
    cupoIngresado >0 && !isNaN(cupoIngresado)) {
        usuarioLocallogueado.cupoMaximo = cupoIngresado;
        mensaje = "El cupo maximo ha sido modificado";
    } else if
    (localTieneReservasPendientesEnEstadoPendiente(usuarioLocallogueado)==0 ||
    isNaN(cupoIngresado)) {
        mensaje = "Debe ingresar un numero mayor a cero";
    } else if
    (localTieneReservasPendientesEnEstadoPendiente(usuarioLocallogueado)>0) {
        mensaje= "Local con reservas pendientes, no puede modificar cupo maximo";
    }
    document.querySelector("#pMensajebtnModificarCuposUsuarioLocal").innerHTML
    = mensaje;
}

```

//FUNCION QUE REvisa SI LOCAL TIENE RESERVAS EN ESTADO PENDIENTE

```

function localTieneReservasPendientesEnEstadoPendiente(usuarioLocal) {
    let contadorReservasPendientesEnLocal=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idLocal == usuarioLocal.idLocal) {
            if (reservaActual.estado == "pendiente") {
                contadorReservasPendientesEnLocal++;
            }
        }
    }
    return contadorReservasPendientesEnLocal;
}

```

//FUNCION QUE MUESTRA PANTALLA CON TABLAS PARA REALIZAR RESERVAS (PERSONA)

```

function btnrealizarReservaHandler(){
    mostrarRealizarReserva();
    TablaReservarLocales();
}

```

//TABLA PARA MOSTRAR LISTADO DE LOCALES PARA RESERVAR

```

function TablaReservarLocales() {
    let bodyHTML = "";
    let textoBoton = "Reservar";
    for (let i=0; i< usuariosLocal.length; i++) {
        let usuarioActual = usuariosLocal[i]
        if (usuarioActual.habilitado == false) {
            bodyHTML += `
            <tr>
                <td>

```

```

        </td>
        <td>
        ${usuarioActual.nombre}
        </td>
        <td>
        <label> Deshabilitado </label>
        </td>
    </tr>
    `;
    }
    else {
        bodyHTML += `
        <tr>
        <td>
        
        </td>
        <td>
        ${usuarioActual.nombre}
        </td>
        <td>
        <input data-localSeleccionado="${usuarioActual.nombreUsuario}"
type="button" value="${textoBoton}" class="btnSeleccionarParaReservar">
        </td>
        </tr>
        `;
    }
    }
    document.querySelector("#bodyTablaReservarLocal").innerHTML = bodyHTML;
    let arrayDeBotones = document.querySelectorAll(".btnSeleccionarParaReservar");
    for (let i=0; i<arrayDeBotones.length; i++) {
        let botonActual = arrayDeBotones[i];
        botonActual.addEventListener("click", btnSeleccionarParaReservar);
    }
}

//FUNCION QUE, LUEGO DE HABER SELECCIONADO LOCAL A RESERVAR,
MUESTRA PANTALLA PARA SELECCIONAR LA CANTIDAD DE CUPOS
function btnSeleccionarParaReservar() {
    mostrarCantidadDeCuposEnRealizarReserva();
    nombreUsuarioLocalSeleccionadoDentroDeRealizarReserva =
this.getAttribute("data-localSeleccionado");
}

```

```

//FUNCION QUE CONFIRMA LA RESERVA LUEGO DE HABER
SELECCIONADO LOS CUPOS
function btnRealizarReservaDentroDeRealizarReservaHandler() {
    let cuposIngresados = parseInt(document.querySelector("#cantidadDeCupos").value);
    let usuarioPersonalogueado = buscarNombreUsuario(usuarioPersona,
nombreUsuarioLogueado);
    let usuarioLocalSeleccionado = buscarNombreUsuario(usuarioLocal,
nombreUsuarioLocalSeleccionadoDentroDeRealizarReserva);
    let totalCuposHastaAhora =
sumarCuposDeUnLocal(usuarioLocalSeleccionado,reservas);
    if (usuarioPersonalogueado !=null &&
personaTieneReservasPendientesEnLocal(usuarioPersonalogueado,
usuarioLocalSeleccionado) == 0) {
        if (isNaN(cuposIngresados) || cuposIngresados == "0" || cuposIngresados == "" ||
cuposIngresados <0) {
            mensaje = "Debe ingresar al menos un cupo";
        } else if (!isNaN(cuposIngresados) && totalCuposHastaAhora + cuposIngresados
> usuarioLocalSeleccionado.cupoMaximo) {
            mensaje = "La cantidad de reservas en estado pendiente es mayor que el cupo
máximo del local. Selecciona menos cupos";
        } else if (!isNaN(cuposIngresados) && totalCuposHastaAhora + cuposIngresados
== usuarioLocalSeleccionado.cupoMaximo) {
            totalCuposHastaAhora = totalCuposHastaAhora + cuposIngresados;
            guardarNuevoUsuarioReservasArray("pendiente", cuposIngresados, 0,
usuarioLocalSeleccionado.idLocal, usuarioPersonalogueado.idPersona);
            mensaje = "La cantidad de reservas totales en estado pendiente coincide con el
cupo máximo del local. Se deshabilitara el local seleccionado";
            usuarioLocalSeleccionado.habilitado = usuarioLocalSeleccionado.habilitado =
false;
        } else {
            totalCuposHastaAhora = totalCuposHastaAhora + cuposIngresados;
            guardarNuevoUsuarioReservasArray("pendiente", cuposIngresados, null,
usuarioLocalSeleccionado.idLocal, usuarioPersonalogueado.idPersona);
            mensaje = "Se ha confirmado la reserva por " + cuposIngresados + " cupos en el
local " + usuarioLocalSeleccionado.nombre;
        }

    } else {
        mensaje = "No se pudo realizar reserva. La persona tiene reservas en estado
pendiente en el local selecccionado";
    }
}

document.querySelector("#pMensajeCantidadDeCuposEnRealizarReserva").innerHTM
L = mensaje;
}

//FUNCION QUE REvisa SI USUARIO PERSONA TIENE RESERVAS
PENDIENTES EN EL LOCAL SELECCIONADO PARA RESERVAR

```

```

function personaTieneReservasPendientesEnLocal(usuarioPersona, usuarioLocal) {
    let contadorReservasPendientes=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idPersona == usuarioPersona.idPersona &&
reservaActual.idLocal == usuarioLocal.idLocal) {
            if (reservaActual.estado == "pendiente") {
                contadorReservasPendientes++;
            }
        }
    }
    return contadorReservasPendientes;
}

```

```

//FUNCION QUE SUMA LOS CUPOS DE RESERVA EN LOCAL
function sumarCuposDeUnLocal(usuarioLocalSeleccionado,reservas) {
    let totalCuposHastaAhora = 0;
    for (let i=0; i< reservas.length; i++) {
        let reservaActual = reservas[i];
        let reservaId = reservaActual.idLocal;
        let cuposReservados = reservaActual.cupos;
        let localId= usuarioLocalSeleccionado.idLocal;
        if (reservaId == localId && reservaActual.estado == "pendiente") {
            totalCuposHastaAhora+= cuposReservados;
        }
    }
    return totalCuposHastaAhora;
}

```

```

//FUNCION QUE MUESTRA PANTALLA PARA CANCELAR RESERVAS
function btnCancelarReservaHandler(){
    mostrarCancelarReserva();
    TablaCancelarReservas();
}

```

```

//FUNCION QUE GENERA TABLA PARA CANCELAR RESERVAS
function TablaCancelarReservas() {
    let usuarioPersonalogueado = buscarNombreUsuario(usuarioPersona,
nombreUsuarioLogueado);
    let bodyHTML = "";
    let textoBoton = "Cancelar";
    for (let i=0; i< reservas.length; i++) {
        let reservaActual = reservas[i];
        for (let i=0; i< usuariosLocal.length; i++) {
            let localActual = usuariosLocal[i];
            if (reservaActual.idPersona == usuarioPersonalogueado.idPersona &&
reservaActual.idLocal == localActual.idLocal) {
                if (reservaActual.estado == "pendiente") {

```

```

        bodyHTML += `
            <tr>
                <td>
                    
                <td>
                    ${localActual.nombre}
                </td>
                <td>
                    ${reservaActual.cupos}
                </td>
                <td>
                    <input data-reserva-persona="${reservaActual.idReserva}"
type="button" value="${textoBoton}" class="btnmodificarEstadoReservaPersona">
                </td>
            </tr>
        `;
    }
}

}

document.querySelector("#bodyTablaCancelarReserva").innerHTML = bodyHTML;
let arrayDeBotones =
document.querySelectorAll(".btnmodificarEstadoReservaPersona");
for (let i=0; i<arrayDeBotones.length; i++) {
    let botonActual = arrayDeBotones[i];
    botonActual.addEventListener("click",
btnmodificarEstadoReservaPersonaHandler);
}
}

//APLICO LA FUNCION DE MODIFICAR EL ESTAOO DE LA RESERVA
(PERSONA) EN CUESTION
function btnmodificarEstadoReservaPersonaHandler() {
    let idReservaParaCambiarEstadoPersona = this.getAttribute("data-reserva-persona");
    modificarEstadoReservaPersona(idReservaParaCambiarEstadoPersona);
    TablaCancelarReservas();
}

//MODIFICO EL ESTADO DE LA RESERVA (PERSONA)
function modificarEstadoReservaPersona(idReservaEnCuestion) {
    reservaEnCuestion = buscarReservaPorId(reservas, idReservaEnCuestion);
    reservaEnCuestion.estado = "cancelada";
}

//FUNCION QUE MUESTRA PANTALLA PARA CAMBIAR RESERVAS DE
ESTADO
function btnCambiarLasReservaDeEstadoHandler() {

```

```

    mostrarCambiarReservaDeEstadoLocal();
    TablaEstadoReservasLocal();
}

//FUNCION QUE INGRESA UN TEXTO Y LLAMA A FUNCION QUE BUSCA
NOMBRE POR SUBCADENA
function btnBuscarNombreSubcadenaHandler() {
    palabraPersonaIngresada= document.querySelector("#buscarPersona").value;

    buscarNombrePersonaParaListado(palabraPersonaIngresada,usuariosPersona,reservas);
    vaciarCampos();
}

//APLICO LA FUNCION DE MODIFICAR EL ESTADO DE LA RESERVA
(LOCAL) EN CUESTION PARA LA TABLA FILTRADA
function btnmodificarEstadoReservaLocalFiltradaHandler() {
    let idReservaParaCambiarEstadoLocal = this.getAttribute("data-reserva-local-
tablaFiltrada");
    modificarEstadoReservaLocal(idReservaParaCambiarEstadoLocal);

    buscarNombrePersonaParaListado(palabraPersonaIngresada,usuariosPersona,reservas);
}

//APLICO LA FUNCION DE MODIFICAR EL ESTADO DE LA RESERVA
(LOCAL) EN CUESTION PARA LA TABLA SIN FILTRAR
function btnmodificarEstadoReservaLocalSinFiltrarHandler() {
    let idReservaParaCambiarEstadoLocal = this.getAttribute("data-reserva-local-
tablaSinFiltrar");
    modificarEstadoReservaLocal(idReservaParaCambiarEstadoLocal);
    TablaEstadoReservasLocal();
}

//MODIFICO EL ESTADO DE LA RESERVA (LOCAL)
function modificarEstadoReservaLocal(idReservaEnCuestion) {
    reservaEnCuestion = buscarReservaPorId(reservas, idReservaEnCuestion);
    reservaEnCuestion.estado = "finalizada";
}

//FUNCION QUE ME RESTABLECE LA TABLA CON TODAS LAS RESERVAS
PENDIENTES (LOCAL)
function btnRestablecerTablaHandler() {
    TablaEstadoReservasLocal();
}

//FUNCION QUE ME DIBUJA LA TABLA CON EL LISTADO DE TODAS LAS
RESERVAS PENDIENTES (LOCAL)
function TablaEstadoReservasLocal() {

```

```

    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);
    let bodyHTML = "";
    let textoBoton = "Finalizar";
    for (let i=0; i< reservas.length; i++) {
        let reservaActual = reservas[i];
        for (let i=0; i< usuariosPersona.length; i++) {
            let usuarioActual = usuariosPersona[i];
            if (reservaActual.idPersona == usuarioActual.idPersona &&
reservaActual.idLocal == usuarioLocallogueado.idLocal) {
                if (reservaActual.estado == "pendiente") {
                    bodyHTML += `
                        <tr>
                            <td>
                                ${usuarioActual.nombre}
                            </td>
                            <td>
                                ${reservaActual.cupos}
                            <td>
                                <input data-reserva-local-
tablaSinFiltrar="${reservaActual.idReserva}" type="button" value="${textoBoton}"
class="btnmodificarEstadoReservaLocalTablaSinFiltrar">
                            </td>
                        </tr>
                    `;
                }
            }
        }
    }

    document.querySelector("#bodyCambiarReservaDeEstadoLocal").innerHTML =
bodyHTML;
    let arrayDeBotones =
document.querySelectorAll(".btnmodificarEstadoReservaLocalTablaSinFiltrar");
    for (let i=0; i<arrayDeBotones.length; i++) {
        let botonActual = arrayDeBotones[i];
        botonActual.addEventListener("click",
btnmodificarEstadoReservaLocalSinFiltrarHandler);
    }
}

////FUNCION QUE BUSCA NOMBRE PERSONA POR SUBCADENA Y DIBUJA
LA TABLA CON EL LISTADO DE LAS CORRESPONDIENTES RESERVAS EN
ESTADO PENDIENTE (LOCAL)
function
buscarNombrePersonaParaListado(palabraPersonaIngresada,usuariosPersona,reservas)
{
    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);

```

```

let bodyHTML = "";
let textoBoton = "Finalizar";
let contador=0;
if (palabraPersonaIngresada == "") {
    document.querySelector("#buscarReservaPorNombre").innerHTML = "Debe
ingresar un texto";
}
else {
    for (let i=0; i< usuariosPersona.length; i++) {
        let usuarioActual = usuariosPersona[i];
        let nombrePersona= usuarioActual.nombre.toLowerCase();
        let subcadena= palabraPersonaIngresada.toLowerCase();
        if (subCadena(nombrePersona,subcadena)==1) {
            contador++;
            for (let i=0; i< reservas.length; i++) {
                let reservaActual = reservas[i];
                if (reservaActual.idPersona == usuarioActual.idPersona &&
reservaActual.idLocal == usuarioLocallogueado.idLocal) {
                    if (reservaActual.estado == "pendiente") {
                        bodyHTML += `
                            <tr>
                                <td>
                                    ${usuarioActual.nombre}
                                </td>
                                <td>
                                    ${reservaActual.cupos}
                                <td>
                                    <input data-reserva-local-
tablaFiltrada="${reservaActual.idReserva}" type="button" value="${textoBoton}"
class="btnmodificarEstadoReservaLocalTablaFiltrada">
                                </td>
                            </tr>
                        `;
                    }
                }
            }
        }
    }
    document.querySelector("#buscarReservaPorNombre").innerHTML = "Se ha/n
encontrado el/los siguiente/s usuario/s: ";
    if (contador ==0) {
        document.querySelector("#buscarReservaPorNombre").innerHTML = "No se
encontro usuario con ese nombre que tenga una reserva pendiente";
    }
}
document.querySelector("#bodyCambiarReservaDeEstadoLocal").innerHTML =
bodyHTML;

```



```

    let arrayDeBotones =
document.querySelectorAll(".btnmodificarEstadoReservaLocalTablaFiltrada");
    for (let i=0; i<arrayDeBotones.length; i++) {
        let botonActual = arrayDeBotones[i];
        botonActual.addEventListener("click",
btnmodificarEstadoReservaLocalFiltradaHandler);
    }
}

//FUNCION QUE BUSCA SI SUBCADENA ESTA CONTENIDA EN LA CADENA
function subCadena(cadena,subcadena) {
    let resultado=-1;
    let i=0;
    let j=0;
    while (i < cadena.length && j<subcadena.length) {
        if (cadena[i]==subcadena[j]) {
            j++;
        } else {
            j=0;
        }
        i++
    }
    if (j == subcadena.length) {
        resultado = 1
    }
    return resultado;
}

//FUNCION QUE MUESTRA LA PANTALLA PARA CALIFICAR RESERVAS
function mostrarCalificarReservas(){
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
    vaciarCampos();
    borrarMensajes();
    document.querySelector("#divCalificarReservasUsuarioPersona").style.display =
"block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA DE
CALIFICAR RESERVAS Y DIBUJA TABLA DE CALIFICAR RESERVAS
function btnCalificarReservaHandler(){
    mostrarCalificarReservas();
    TablaCalificarReservas();
    borrarMensajes();
}

```

```

//BOTON QUE LLAMA FUNCION PARA CALIFICAR LA RESERVA
function btnCalificarReservaDentroDeCalificarReservaHandler() {
    let idReservaParaCalificar = this.getAttribute("data-reserva-local-a-calificar");
    calificarReserva(idReservaParaCalificar);
    TablaCalificarReservas();
}

//CALIFICO LA RESERVA
function calificarReserva(idReservaEnCuestion) {
    let reservaEnCuestion = buscarReservaPorId(reservas, idReservaEnCuestion);
    calificacion = parseInt(document.querySelector("#comboCalificarReservas" +
    reservaEnCuestion.idReserva).value);
    if (!isNaN(calificacion)) {
        reservaEnCuestion.calificacion = calificacion;
        document.querySelector("#pCalificarReserva").innerHTML = "La reserva ha sido
        calificada con exito";
    } else {
        document.querySelector("#pCalificarReserva").innerHTML = "Debe seleccionar
        una calificacion del 1 al 5";
    }
}

function TablaCalificarReservas() {
    let usuarioPersonalogueado = buscarNombreUsuario(usuariosPersona,
    nombreUsuarioLogueado);
    let bodyHTML = "";
    let textoBoton = "Calificar";
    for (let i=0; i< reservas.length; i++) {
        let reservaActual = reservas[i];
        for (let i=0; i< usuariosLocal.length; i++) {
            let localActual = usuariosLocal[i];
            if (reservaActual.idPersona == usuarioPersonalogueado.idPersona &&
            reservaActual.idLocal == localActual.idLocal) {
                if (reservaActual.estado == "finalizada") {
                    bodyHTML += `
                        <tr>
                            <td>
                                ${localActual.nombre}
                            </td>
                            <td>
                                ${reservaActual.cupos}
                            </td>
                            <td>
                                <select
id="comboCalificarReservas${reservaActual.idReserva}">
                                    <option value="">Calificar</option>
                                    <option value="1">1</option>
                                    <option value="2">2</option>

```

```

                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option></select>
            </td>
            <td>
                <input data-reserva-local-a-
calificar="{reservaActual.idReserva}" type="button" value="{textoBoton}"
class="btnCalificarReservaDentroDeCalificarReserva">
            </td>
        </tr>
    `;
    }
    }
    }
}

```

```

document.querySelector("#bodyTablaCalificarReservasUsuarioPersona").innerHTML =
bodyHTML;

```

```

    let arrayDeBotones =
document.querySelectorAll(".btnCalificarReservaDentroDeCalificarReserva");
    for (let i=0; i<arrayDeBotones.length; i++) {
        let botonActual = arrayDeBotones[i];
        botonActual.addEventListener("click",
btnCalificarReservaDentroDeCalificarReservaHandler);
    }
}

```

```

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA DE VER
RESERVAS PENDIENTES Y DIBUJA TABLA DE RESERVAS PENDIENTES
function btnVerMisReservasPendientesHandler() {
    mostrarVerMisReservasPendientes();
    TablaVerReservasPendientes();
}

```

```

//FUNCION QUE MUESTRA LA PANTALLA PARA VER RESERVAS
PENDIENTES
function mostrarVerMisReservasPendientes() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
    document.querySelector("#divVerReservaPendientesUsuarioPersona").style.display
= "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

```

```

//FUNCION QUE DIBUJA TABLA PARA VER RESERVAS PENDIENTES
function TablaVerReservasPendientes() {
    let usuarioPersonalogueado = buscarNombreUsuario(usuarioPersona,
nombreUsuarioLogueado);
    let bodyHTML = "";

    for (let i=0; i< reservas.length; i++) {
        let reservaActual = reservas[i];
        for (let i=0; i< usuariosLocal.length; i++) {
            let localActual = usuariosLocal[i];
            if (reservaActual.idPersona == usuarioPersonalogueado.idPersona &&
reservaActual.idLocal == localActual.idLocal) {
                if (reservaActual.estado == "pendiente") {
                    bodyHTML += `
                        <tr>
                            <td>
                                
                                </td>
                            <td>
                                ${localActual.nombre}
                            </td>
                            <td>
                                ${reservaActual.cupos}
                            </td>
                        </tr>
                    `;
                }
            }
        }
    }

    document.querySelector("#bodyVerReservaPendientesUsuarioPersona").innerHTML =
bodyHTML;
}

//FUNCION QUE MUESTRA LA PANTALLA VISUALIZAR INFORMACION
ESTADISTICA LOCAL
function mostrarInformacionEstadisticaLocal() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();
    document.querySelector("#divInformacionEstadisticaLocal").style.display = "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA
VISUALIZAR INFORMACION ESTADISTICA LOCAL

```

```

function btnInformacionEstadisticaLocalHandler() {
    mostrarInformacionEstadisticaLocal();
}

//FUNCION QUE MUESTRA LA PANTALLA "VER PORCENTAJE DE
OCUPACION DEL LOCAL"
function mostrarverElPorcentajeDeOcupacionDelLocal() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();
    document.querySelector("#divverElPorcentajeDeOcupacionDelLocal").style.display
= "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA "VER
PORCENTAJE DE OCUPACION DEL LOCAL"
function btnverElPorcentajeDeOcupacionDelLocalHandler() {
    mostrarverElPorcentajeDeOcupacionDelLocal();
    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);
    porcentaje = calcularPorcentajeDeOcupacionDelLocal(usuarioLocallogueado);
    document.querySelector("#pverElPorcentajeDeOcupacionDelLocal").innerHTML =
"El porcentaje de ocupacion del local es " + porcentaje + "%";
}

//FUNCION QUE CALCULA EL PORCENTAJE DE OCUPACION DEL LOCAL
function calcularPorcentajeDeOcupacionDelLocal(usuarioLocal) {
    let cuposLocal = sumarCuposDeUnLocal(usuarioLocal,reservas)
    let porcentaje = (100 * cuposLocal / usuarioLocal.cupoMaximo).toFixed(2);
    return porcentaje;
}

//FUNCION QUE MUESTRA LA PANTALLA "VER PROMEDIO DE
CALIFICACIONES DEL LOCAL"
function mostrarerElPromedioDeCalificacionesDelLocal() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();

    document.querySelector("#divverElPromedioDeCalificacionesDelLocal").style.display
= "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA "VER
PROMEDIO DE CALIFICACIONES DEL LOCAL"
function btnverElPromedioDeCalificacionesDelLocalHandler() {

```

```

    mostrarElPromedioDeCalificacionesDelLocal();
    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);
    promedio = calcularPromedioDeCalificacionesDelLocal(usuarioLocallogueado);
    if (isNaN(promedio)) {

document.querySelector("#pverElPromedioDeCalificacionesDelLocal").innerHTML =
"El local todavia no cuenta con reservas calificadas";
    }
    else {

document.querySelector("#pverElPromedioDeCalificacionesDelLocal").innerHTML =
"El promedio de calificaciones del local es " + promedio;
    }
}

//FUNCION QUE CALCULA EL PROMEDIO DE CALIFICACIONES DEL LOCAL
function calcularPromedioDeCalificacionesDelLocal(usuarioLocal) {
    let sumaCalificaciones = 0;
    let contadorCalificaciones = 0;
    for (let i=0; i < reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idLocal == usuarioLocal.idLocal && reservaActual.estado ==
"finalizada" && reservaActual.calificacion != null) {
            sumaCalificaciones+= reservaActual.calificacion;
            contadorCalificaciones ++;
        }
    }
    let promedio = (sumaCalificaciones / contadorCalificaciones).toFixed(2);
    return promedio;
}

//FUNCION QUE MUESTRA LA PANTALLA "VER LISTADO DE TODOS LOS
LOCALES Y PROMEDIO DE SUS CALIFICACIONES"
function
mostrarListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUno() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();

document.querySelector("#divverListadoDeTodosLosLocalesYElPromedioDeCalificaci
onesDeCadaUno").style.display = "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA "VER
LISTADO DE TODOS LOS LOCALES Y PROMEDIO DE SUS CALIFICACIONES"

```

```
function
btnverListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUnoHandler()
{
    mostrarListadoDeTodosLosLocalesYElPromedioDeCalificacionesDeCadaUno();
    TablaReservasConSuPromedioDeCalificaciones();
}
```

//FUNCION QUE CALCULA EL PROMEDIO DE CALIFICACIONES DEL LOCAL

```
function TablaReservasConSuPromedioDeCalificaciones() {
    let bodyHTML = "";
    for (let i=0; i< usuariosLocal.length; i++) {
        let localActual = usuariosLocal[i];
        promedio = calcularPromedioDeCalificacionesDelLocal(localActual);
        if (isNaN(promedio)) {
            promedio = "El local todavia no cuenta con reservas calificadas";
        }
        bodyHTML += `
            <tr>
            <td>
                
            </td>
            <td>
                ${localActual.nombre}
            </td>
            <td>
                ${promedio}
            </td>
            </tr>
        `;
    }
}
```

```
document.querySelector("#bodyverListadoDeTodosLosLocalesYElPromedioDeCalifica
cionesDeCadaUno").innerHTML = bodyHTML;
}
```

//FUNCION QUE MUESTRA LA PANTALLA "VER TOTAL DE RESERVAS REALIZADAS HASTA EL MOMENTO"

```
function mostrarVerElTotalDeReservasRealizadasHastaElMomento() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverLocal();
}
```

```
document.querySelector("#divVerElTotalDeReservasRealizadasHastaElMomento").styl
e.display = "block";
document.querySelector("#btnVolverInicio").style.display = "none";
}
```

```

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA "VER
TOTAL DE RESERVAS REALIZADAS HASTA EL MOMENTO"
function btnVerElTotalDeReservasRealizadasHastaElMomentoHandler() {
    mostrarVerElTotalDeReservasRealizadasHastaElMomento();
    let usuarioLocallogueado = buscarNombreUsuario(usuariosLocal,
nombreUsuarioLogueado);
    contadorReservasPendientes =
localTieneReservasPendientesEnEstadoPendiente(usuarioLocallogueado);
    contadorReservasFinalizadas =
cuentaReservasFinalizadasEnLocal(usuarioLocallogueado);

    document.querySelector("#pVerElTotalDeReservasRealizadasHastaElMomento").innerHTML = "El local tiene " + contadorReservasPendientes + " reservas pendientes y " +
contadorReservasFinalizadas + " reservas finalizadas";
}

//FUNCION QUE CUENTA LA CANTIDAD DE RESERVAS FINALIZADAS EN
LOCAL INGRESADO
function cuentaReservasFinalizadasEnLocal(usuarioLocal) {
    let contadorReservasFinalizadas=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idLocal == usuarioLocal.idLocal) {
            if (reservaActual.estado == "finalizada") {
                contadorReservasFinalizadas++;
            }
        }
    }
    return contadorReservasFinalizadas;
}

//FUNCION QUE CUENTA LA CANTIDAD DE RESERVAS TOTALES EN
LOCAL
function cuentaReservasEnLocal(usuarioLocal) {
    let contadorReservas=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idLocal == usuarioLocal.idLocal) {
            contadorReservas++;
        }
    }
    return contadorReservas;
}

function mostrarInformacionEstadisticaPersona() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
}

```



```

    document.querySelector("#divInformacionEstadisticaPersona").style.display =
    "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
}

//BOTON QUE LLAMA A FUNCION PARA MOSTRAR LA PANTALLA
VISUALIZAR INFORMACION ESTADISTICA PERSONA
function btnInformacionEstadisticaPersonaHandler() {
    mostrarInformacionEstadisticaPersona();
}

//FUNCION PARA VER TABLA DE PORCENTAJES DE RESERVAS PERSONA
function mostrarListadoReservasUsuarioPersona() {
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
    document.querySelector("#divListadoReservasUsuarioPersona").style.display =
    "block";
    document.querySelector("#btnVolverInicio").style.display = "none";
    TablaListadoReservasInformacionEstadisticaPersona();
}

//BOTON QUE LLAMA A FUNCION PARA VER TABLA DE PORCENTAJES DE
RESERVAS PERSONA
function btnverElPorcentajeDeReservasHandler(){
    mostrarListadoReservasUsuarioPersona();
}

// TABLA MUESTRA INFORMACION Y PORCENTAJE DE RESERVAS DE UNA
PERSONA. SE IMPRIME LA CANTIDAD DE RESERVAS QUE HAY EN ESE
LUGAR. HAY QUE VER UNA MANERA DE HACER QUE SOLO IMPRIMA
CUANDO ENTRE UNA VEZ A LA RESERVA Y TA.
function TablaListadoReservasInformacionEstadisticaPersona() {
    let usuarioPersonalogueado = buscarNombreUsuario(usuarioPersona,
nombreUsuarioLogueado);
    let totalReservasDelLocal=0
    let porcentaje=0;
    let bodyHTML = "";
    for (let j=0; j< usuariosLocal.length; j++) {
        let localActual = usuariosLocal[j];
        let cantidadReservas =
cuentaReservasFinalizadasPorUsuarioPersona(usuarioPersonalogueado, localActual);
        totalReservasDelLocal = cuentaReservasEnLocal(localActual);
        porcentaje= ((cantidadReservas*100)/totalReservasDelLocal).toFixed(2);
        if (cantidadReservas >0){
            bodyHTML += `
                <tr>
                <td>

```

```

        ${localActual.nombre}
    </td>
    <td>
        ${cantidadReservas}
    </td>
    <td>
        ${totalReservasDelLocal}
    </td>
    <td>
        ${porcentaje}%
    </td>
</tr>
`;
    }
}
document.querySelector("#bodyListadoReservasUsuarioPersona").innerHTML =
bodyHTML;
}

//FUNCION QUE CUENTA LA CANTIDAD DE RESERVAS FINALIZADAS
HECHAS POR UN USUARIO PERSONA
function cuentaReservasFinalizadasPorUsuarioPersona(usuarioPersona, usuarioLocal) {
    let contadorReservasHechas=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idPersona == usuarioPersona.idPersona &&
reservaActual.idLocal == usuarioLocal.idLocal && reservaActual.estado ==
"finalizada") {
            contadorReservasHechas++;
        }
    }
    return contadorReservasHechas;
}

//FUNCION QUE CUENTA LA CANTIDAD DE RESERVAS PENDIENTES
HECHAS POR UN USUARIO PERSONA
function cuentaReservasPendientesPorUsuarioPersona(nombreUsuarioPersona,
nombreUsuarioLocal) {
    usuarioPersona = buscarNombreUsuario(usuarioPersona, nombreUsuarioPersona);
    usuarioLocal = buscarNombreUsuario(usuarioLocal, nombreUsuarioLocal);
    let contadorReservasHechas=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idPersona == usuarioPersona.idPersona &&
reservaActual.idLocal == usuarioLocal.idLocal && reservaActual.estado ==
"pendiente") {

```

```

        contadorReservasHechas++;
    }
}
return contadorReservasHechas;
}

```

//FUNCION QUE CUENTA LA CANTIDAD DE RESERVAS TOTALES HECHAS POR UN USUARIO PERSONA

```

function cuentaReservasPorUsuarioPersona(usuarioPersona, usuarioLocal) {
    let contadorReservasHechas=0;
    for (let i=0; i<reservas.length; i++) {
        let reservaActual = reservas[i];
        if (reservaActual.idPersona == usuarioPersona.idPersona &&
            reservaActual.idLocal == usuarioLocal.idLocal) {
            contadorReservasHechas++;
        }
    }
    return contadorReservasHechas;
}

```

//FUNCION PARA VER LISTADO DE LOCAL DONDE HA HECHO MAS RESERVAS

```

function mostrarLocalDondeHaHechoMasReservas(){
    ocultarPantallas();
    mostrarCerrarSesion();
    mostrarVolverPersona();
}

```

```

document.querySelector("#divverElLocalDondeHaHechoMasReservas").style.display =
"block";
document.querySelector("#btnVolverInicio").style.display = "none";
TablaListadoLocalDondeHaHechoMasReservas();
}

```

//BOTON QUE LLAMA A FUNCION PARA VER TABLA DE PORCENTAJES DE RESERVAS PERSONA

```

function btnverElLocalDondeHaHechoMasReservasHandler(){
    mostrarLocalDondeHaHechoMasReservas();
}

```

// TABLA MUESTRA LOCAL O LOCALES DONDE HA HECHO MAS RESERVAS. TAMPOCO FUNCIONA DEL TODO BIEN

```

function TablaListadoLocalDondeHaHechoMasReservas() {
    let localesConMaximaCantidadDeReservasEncontradas = [];
    let maximaCantidadDeReservasEncontradas = Number.NEGATIVE_INFINITY;
    let usuarioPersonalogueado = buscarNombreUsuario(usuarioPersona,
nombreUsuarioLogueado);
}

```

```

    let bodyHTML = "";
    for (let i=0; i< usuariosLocal.length; i++) {
        let localActual = usuariosLocal[i];
        let cant =
cuentaReservasPorUsuarioPersona(usuarioPersonalogueado,localActual);
        if (cant > maximaCantidadDeReservasEncontradas) {
            maximaCantidadDeReservasEncontradas = cant;
            localesConMaximaCantidadDeReservasEncontradas = [];

localesConMaximaCantidadDeReservasEncontradas.push(localActual.nombre);
        }
        else if (cant == maximaCantidadDeReservasEncontradas) {

localesConMaximaCantidadDeReservasEncontradas.push(localActual.nombre);
        }
    }
    for (let i=0; i< localesConMaximaCantidadDeReservasEncontradas.length; i++)
    {
        let nombreLocalActual = localesConMaximaCantidadDeReservasEncontradas
[i];
        if (maximaCantidadDeReservasEncontradas > 0) {
            bodyHTML += `
                <tr>
                <td>
                    ${nombreLocalActual}
                </td>
                <td>
                    ${maximaCantidadDeReservasEncontradas}
                </td>
                </tr>
            `;
        }
    }

document.querySelector("#bodyverElLocalDondeHaHechoMasReservas").innerHTML
= bodyHTML;
}

```