

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 1
DOCUMENTO DE ANÁLISIS



[Nicolás Damelles Martinez – 286372]



[Marcos Parma Sanchez – 285393]

M1D

Docente: Sergio Palay

Analista en Tecnologías de la Información

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	6
2.1. F01-Inicio de Sesión	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 – Registro Usuario	8
2.2.1. Acceso	8
2.2.2. Descripción	8
2.2.3. Interfaz de usuario	8
2.2.4. Validaciones	8
2.3. F03 – Realizar la solicitud de una Reserva	9
2.3.1. Acceso	9
2.3.2. Descripción	9
2.3.3. Interfaz de usuario	9
2.3.4. Validaciones	9
2.4. F04 – Cancelar una Reserva	10
2.4.1. Acceso	10
2.4.2. Descripción	10
2.4.3. Interfaz de usuario	10
2.4.4. Validaciones	10

2.5. F05 – Estadísticas	11
2.5.1. Acceso	11
2.5.2. Descripción	11
2.5.3. Interfaz de usuario	11
2.5.4. Validaciones	11
2.6. F06 – Calificación del Local	12
2.6.1. Acceso	12
2.6.2. Descripción	12
2.6.3. Interfaz de usuario	12
2.6.4. Validaciones	12
2.7. F07 – Deshabilitar disponibilidad	13
2.7.1. Acceso	13
2.7.2. Descripción	13
2.7.3. Interfaz de usuario	13
2.7.4. Validaciones	13
2.8. F08 – Cambiar las reservas de estado	14
2.8.1. Acceso	14
2.8.2. Descripción	14
2.8.3. Interfaz de usuario	14
2.8.4. Validaciones	14
2.9. F09 – Modificar cupo máximo	15
2.9.1. Acceso	15
2.9.2. Descripción	15
2.9.3. Interfaz de usuario	15
2.9.4. Validaciones	15
2.10. F010 – Estadísticas Local	16
2.10.1. Acceso	16
2.10.2. Descripción	16
2.10.3. Interfaz de usuario	16

2.10.4. Validaciones	16
3. INFORME TESTING	17 a 21
4. Código HTML	22 a 38
5. Código JAVASCRIPT	39 a 94
6. LISTA INFORMACIÓN PRECARGADA	95

1. Descripción general del problema a resolver

Se solicita una aplicación que permita gestionar las reservas de locales junto con otras funciones.

1.1. Tipos de usuario del sistema

Hay dos tipos de usuarios, el tipo “Local” (de tipo administrador), y tipo “Persona” (usuario)

1.2. Listado de funcionalidades

[Se debe presentar una lista de las funcionalidades del obligatorio]

[Cada una de las funcionalidades deberá incluir: Numero – Titulo – Usuarios que pueden acceder]

[Las funcionalidades deberán estar ordenadas por tipo de usuario.]

- **F01 – Inicio de Sesión– Persona y Local**
- **F02 – Registro Usuario – Persona**
- **F03 – Realizar la solicitud de una Reserva– Persona**
- **F04 – Cancelar una Reserva– Persona**
- **F05 – Estadísticas– Persona**
- **F06 – Calificación del Local– Persona**
- **F07 – Deshabilitar disponibilidad– Local**
- **F08 – Cambiar las reservas de estado–Local**
- **F09 – Modificar cupo máximo–Local**
- **F010 – Estadísticas Local–Local**

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-Inicio de Sesión

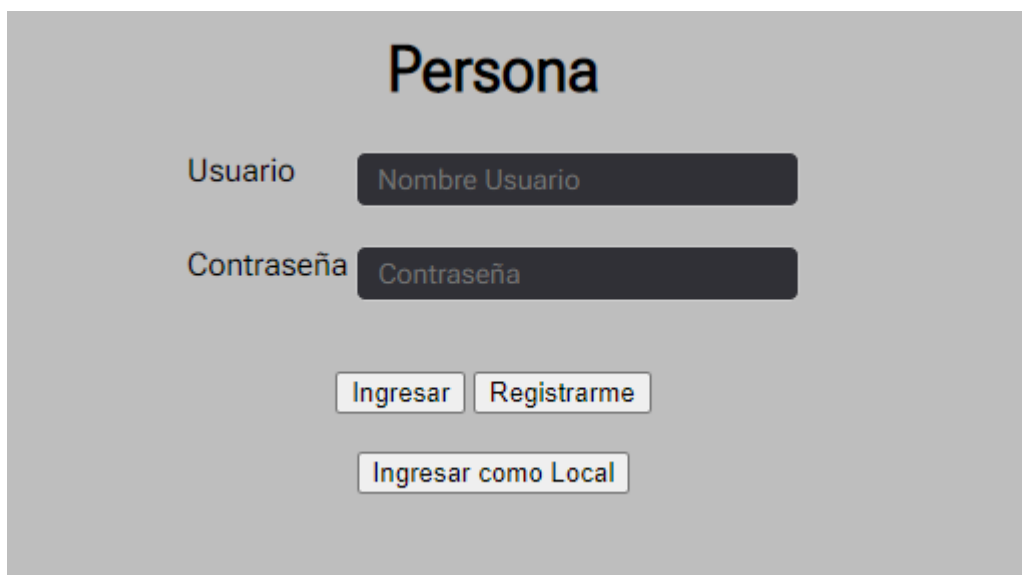
2.1.1. Acceso

Persona y Local

2.1.2. Descripción

Validar si el usuario y contraseña ingresado pertenece a una persona o local registrado previamente.

2.1.3. Interfaz de usuario



La interfaz de usuario para el inicio de sesión de una persona se muestra en un recuadro gris. En la parte superior, el título "Persona" está en un tamaño de fuente grande y color negro. Debajo del título, hay dos campos de entrada de texto: "Usuario" y "Contraseña". El campo "Usuario" tiene un placeholder "Nombre Usuario" y el campo "Contraseña" tiene un placeholder "Contraseña". Debajo de los campos, hay tres botones: "Ingresar", "Registrarme" y "Ingresar como Local". El botón "Ingresar" y "Registrarme" están juntos y el botón "Ingresar como Local" está debajo de ellos.

2.1.4. Validaciones

- **Usuario:** Se validará que el nombre de usuario haya sido ingresado previamente
 - Mensaje de Error: "Nombre de usuario invalido o inexistente"
- **Clave:** Se validará que la clave ingresada pertenezca y coincida con el usuario
 - Mensaje de Error: "Contraseña Invalida"
- **Campos Obligatorios:** Todos.

2.2. F02 – Registro Usuario

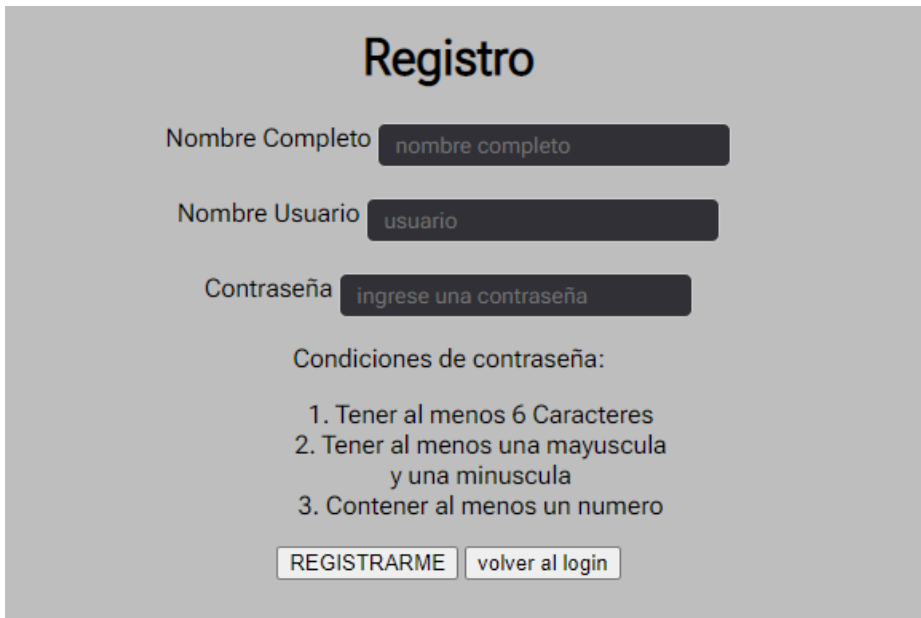
2.2.1. Acceso

Persona

2.2.2. Descripción

Formulario de registro donde se solicitan todos los datos necesarios para ingresar un nuevo Usuario (Persona)

2.2.3. Interfaz de usuario



The screenshot shows a registration form titled "Registro". It contains three input fields: "Nombre Completo" with placeholder text "nombre completo", "Nombre Usuario" with placeholder text "usuario", and "Contraseña" with placeholder text "ingrese una contraseña". Below the password field, there is a section titled "Condiciones de contraseña:" followed by three numbered requirements: "1. Tener al menos 6 Caracteres", "2. Tener al menos una mayúscula y una minúscula", and "3. Contener al menos un numero". At the bottom of the form, there are two buttons: "REGISTRARME" and "volver al login".

2.2.4. Validaciones

- **Nombre Usuario:** Se validará que el nombre de usuario no coincida con algún otro usuario ingresado previamente.
 - Mensaje de Error: “Nombre de Usuario ya existente”
- **Contraseña:** Se validará que la contraseña ingresada contenga los siguientes parámetros: un mínimo de 6 caracteres, contando con al menos una mayúscula, una minúscula y un número.
 - Mensaje de Error: “Clave Ingresada, Invalida”
- **Campos Obligatorios:** Todos.

2.3. F03 – Realizar la solicitud de una Reserva

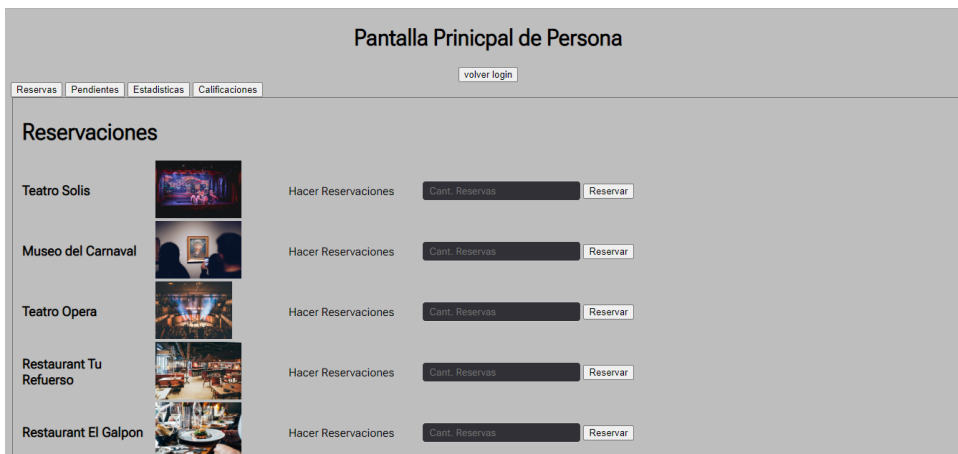
2.3.1. Acceso

Persona

2.3.2. Descripción

Permite al usuario realizar una reserva a un local, ingresando la cantidad de lugares a reservar.

2.3.3. Interfaz de usuario



2.3.4. Validaciones

- **Reserva pendiente en local:** Se validará que el usuario no tenga una reserva “Pendiente” en el local que intenta reservar.
 - Mensaje de Error: “Ya hay una reserva Pendiente”
- **Local habilitado:** Se validará que el local esté habilitado para que los usuarios puedan hacer reservas.
 - Mensaje de Error: “El local está deshabilitado momentáneamente”
- **Cupos Autorizados:** Se validará que el local cuente con cupos disponibles para hacer la reserva.
 - Mensaje de Error: “Local lleno”
- **Campos Obligatorios:** Todos.

2.4. F04 – Cancelar una Reserva

2.4.1. Acceso

Persona

2.4.2. Descripción

El usuario podrá cancelar una reserva “pendiente” propia, pasando la misma a estado “cancelada”, solamente se mostrarán las reservas con estado “pendiente”

2.4.3. Interfaz de usuario

The screenshot shows a web interface titled "Pantalla Principal de Persona". At the top, there are four tabs: "Reservas", "Pendientes", "Estadísticas", and "Calificaciones". A "volver login" button is located to the right of these tabs. Below the tabs, the section "Reservas Pendientes" is displayed. It contains a table with the following data:

Usuario	Nombre Local	Lugares Reservados	Estado Reserva	
marcos	Teatro Solis	23	Pendiente	Cancelar Reserva
marcos	Museo del Carnaval	23	Pendiente	Cancelar Reserva
marcos	Teatro Opera	23	Pendiente	Cancelar Reserva
marcos	Tu Refuerso	33	Pendiente	Cancelar Reserva

2.4.4. Validaciones

- **Reserva pendiente en local:** Se validará que el usuario tenga reservas con estado “pendiente”.

2.5. F05 – Estadísticas

2.5.1. Acceso

Persona

2.5.2. Descripción

El usuario podrá ver un listado de los locales donde tenga reservas completadas, la cantidad de reservas realizadas, la cantidad total de reservas que tiene ese local y que porcentaje de estas pertenecen al usuario.

El usuario podrá ver en qué local realizó más reservas.

2.5.3. Interfaz de usuario

Pantalla Principal de Persona

Reservas | Pendientes | Estadísticas | Calificaciones volver login

Pantalla ESTADISTICAS Usuario Persona

Locales	Cantidad	Total Reservas	%Reserva
Teatro Solis	2	2	%100.0
Museo del Carnaval	1	1	%100.0
Teatro Opera	1	1	%100.0
Tu Refuerso	1	1	%100.0

Locales mas reservados por el usuario:
Teatro Solis

2.5.4. Validaciones

- **Reservas finalizadas:** Se validará que el usuario tenga reservas con estado “finalizada”.
 - Mensaje de Error: “ N /A ”

2.6. F06 – Calificación del Local

2.6.1. Acceso

Persona

2.6.2. Descripción

El usuario podrá calificar la reservas “finalizadas” y puntuarla del 1 al 5 según la experiencia que haya tenido,

2.6.3. Interfaz de usuario

The screenshot shows the 'Pantalla Principal de Persona' interface. At the top, there are tabs for 'Reservas', 'Pendientes', 'Estadísticas', and 'Calificaciones', with a 'volver login' button. Below the tabs, the title 'Pantalla Reservas Usuario Persona' is displayed. A table lists reservations for the user 'marcos'. Each row includes the local name, number of reserved seats, reservation status, a rating dropdown menu, and a 'Calificar Reserva' button.

Usuario	Nombre Local	Lugares Reservados	Estado Reserva		
marcos	Teatro Solis	23	Finalizada	3	Calificar Reserva
marcos	Museo del Carnaval	23	Finalizada	5	Calificar Reserva
marcos	Teatro Opera	23	Finalizada	3	Calificar Reserva
marcos	Tu Refuerzo	33	Finalizada	1	Calificar Reserva
marcos	Teatro Solis	23	Finalizada	4	Calificar Reserva

2.6.4. Validaciones

- **Reservas finalizadas:** Se validará que el usuario tenga reservas con estado “finalizada”.
 - Mensaje de Error: N/A

2.7. F07 – Deshabilitar disponibilidad

2.7.1. Acceso

Local

2.7.2. Descripción

El local podrá habilitar o deshabilitar la posibilidad de que los usuarios hagan reservas, se deshabilita esta posibilidad automáticamente, si el local llega a su máximo de cupos.

2.7.3. Interfaz de usuario

The screenshot displays the 'Pantalla Principal Local' (Main Local Screen) interface. At the top, there's a header with the title 'Pantalla Principal Local' and a 'Volver login' button. Below the header, there are three tabs: 'Informacion Local', 'Estadísticas', and 'Gestion Reservas'. The 'Informacion Local' tab is active, showing 'Pantalla INFO LOCAL Usuario Local'. The main content area is divided into two sections. The left section, titled 'Estado del local:', shows the current state of the local as 'Teatro Solis' with a dropdown menu set to 'Habilitado' and a 'Cambiar' button. Below this, it says 'Estado Actual:habilitado' and 'Mensaje Cambio Estado'. The right section, titled 'Ingrese Nuevo Cupo Maximo', shows a text input field with '100' and a 'Cambiar' button. Below this, it says 'Cambio de cupos realizado' and 'Cupo Max Actual:100'.

2.7.4. Validaciones

- **Actuales Reservas:** Se validará que el local no tenga reservas pendientes
 - Mensaje de Error: “El local tiene reservas pendientes”

2.8. F08 – Cambiar las reservas de estado

2.8.1. Acceso

Local

2.8.2. Descripción

El local podrá buscar por nombre de usuario entre las reservas “pendientes” para cambiar el estado de una reserva puntual de “pendiente” a “finalizada”.

2.8.3. Interfaz de usuario

Pantalla Principal Local

Volver login

Informacion Local Estadísticas Gestion Reservas

Pantalla GESTION RESERVAS Usuario Local

Buscar Pendientes Buscar

Mostrar Todas

Usuario	Nombre Local	Lugares Reservados	Estado Reserva	
marcos	Teatro Solis	23	Pendiente	Finalizar Reserva
nicolas	Teatro Solis	12	Pendiente	Finalizar Reserva
marcos	Teatro Solis	23	Finalizada	
marcos	Teatro Solis	23	Finalizada	

2.8.4. Validaciones

- **Nombre Existente:** Se validará que el usuario ya exista al hacer la búsqueda
 - Mensaje de Error: “No se encontraron resultados”

2.9. F09 – Modificar cupo máximo

2.9.1. Acceso

Local

2.9.2. Descripción

El local podrá modificar su cupo máximo, esta opción solo estará disponible si el local no tiene reservas en estado “pendiente”.

2.9.3. Interfaz de usuario

The screenshot displays the 'Pantalla Principal Local' interface. At the top, there are tabs for 'Informacion Local', 'Estadisticas', and 'Gestion Reservas', along with a 'Volver login' button. The main section is titled 'Pantalla INFO LOCAL Usuario Local'. It features a form for modifying the local's status and capacity. On the left, the local name 'Teatro Solis' is displayed. Next to it is a dropdown menu for 'Estado del local' currently set to 'Habilitado', followed by a 'Cambiar' button. Below this is a 'Mensaje Cambio Estado' label. To the right, there is a text input field for 'Ingrese Nuevo Cupo Maximo' with the value '100', a 'Cambiar' button, and a 'Cambio de cupos realizado' label. At the bottom right, it shows 'Cupo Max Actual:100'.

2.9.4. Validaciones

- **Reservas Actuales:** Se validará que el local no tenga reservas pendientes, en dicho caso no se podrá modificar el cupo del local.
 - Mensaje de Error: “Aun hay reservas en estado pendiente, no se puede modificar el cupo”

2.10. F010 – Estadísticas Local

2.10.1. Acceso

Local

2.10.2. Descripción

El usuario "Local" ver el porcentaje de ocupación del local, ver el promedio de calificaciones de su local y también un listado de todos los locales y el promedio de calificaciones de cada uno, y podrá ver el total de reservas realizadas hasta el momento.

2.10.3. Interfaz de usuario

% de ocupacion	Promedio Calificaciones	Total Reservas
%50	4.0	
Cupo Maximo:100		Pendientes: 50
Actuales: 50		Finalizadas 20

2.10.4. Validaciones

- **N / A:** -
 - Mensaje de Error: “-”

3. INFORME TESTING

F-01: Iniciar Sesión (Persona / Local).

Precondiciones: - El usuario y contraseña deben coincidir con la información cargada en el Array.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F01-T01	Inicio de sesión correcto	1. Ingresar nombre de usuario 2. Ingresar Contraseña	Array de personas Array de Locales precargados	Mostrar interfaz de Usuario / Local	Mostrar interfaz de Usuario / Local	P
F01-T02	Inicio de Sesión Incorrecto Usuario Inexistente	1. Ingresar nombre de usuario 2. Ingresar Contraseña	Array de personas Array de Locales precargados	Mostrar Mensaje Error: "Nombre de usuario invalido o inexistente"	Mostrar Mensaje Error: "Nombre de usuario invalido o inexistente"	F
F01-T03	Inicio de Sesión Incorrecto Contraseña Invalida	1. Ingresar nombre de usuario 2. Ingresar Contraseña	Array de personas Array de Locales precargados	Mostrar Mensaje Error: "Contraseña Invalida"	Mostrar Mensaje Error: "Contraseña Invalida"	F

F-02: Registrar Usuario.

Precondiciones: -Al registrarse no se puede repetir el nombre de usuario y la contraseña debe cumplir con los parámetros establecidos.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F02-T01	Registro de Nuevo Usuario Correcto	1. Selecciona opción de Registrarse. 2. Ingresa nombre Completo. 3. Ingresa Nombre Usuario. 4. Ingresa Contraseña	Array de Personas	Se agrega nuevo usuario al array de personas. Se muestra mensaje de éxito.	Se agrega nuevo usuario al array de personas. Se muestra mensaje de éxito: "Felicidades te registraste correctamente!, vuelve a login para ingresar."	P
F02-T02	Registro de Nuevo Usuario, Usuario existente	1. Selecciona opción de Registrarse. 2. Ingresa nombre Completo. 3. Ingresa Nombre Usuario. 4. Ingresa Contraseña	Array de Personas Verificación de usuarios existentes en el array	Se muestra mensaje de error. "Nombre de Usuario ya Existente"	Se muestra mensaje de error. "Nombre de Usuario ya Existente"	F
F02-T03	Registro de Nuevo Usuario, Contraseña incorrecta	1. Selecciona opción de Registrarse. 2. Ingresa nombre Completo. 3. Ingresa Nombre Usuario. 4. Ingresa Contraseña	Array Persona Verificación que se cumplan los parámetros establecidos.	Se muestra mensaje de error. "Clave Ingresada Invalida"	Se muestra mensaje de error. "Clave Ingresada Invalida"	F

F-03: Realizar Solicitud de Reserva (Usuario Persona).

Precondiciones: -Se validará que el Usuario no tenga reservas pendientes para ese local, que el local este Habilitado y que el local cuenta con cupos disponibles para hacer la reserva.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F03-T01	Solicitud de Reserva Correcta	1. Ingresar cantidad de cupos en el local. 2. Presionar Botón "Reservar"	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar Mensaje de éxito: "Reserva Exitosa"	Mostrar Mensaje de éxito: "Reserva Exitosa"	P
F03-T02	Solicitud de Reserva Incorrecta Reserva en estado "Pendiente"	1. Ingresar cantidad de cupos en el local. 2. Presionar Botón "Reservar"	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar mensaje de Error: "Ya hay una reserva Pendiente"	Mostrar mensaje de Error: "Ya hay una reserva Pendiente"	F
F03-T03	Solicitud de Reserva Incorrecta Estado del local "Deshabilitado"	1. Ingresar cantidad de cupos en el local. 2. Presionar Botón "Reservar"	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar mensaje de Error: "El local esta Deshabilitado momentáneamente"	Mostrar mensaje de Error: "El local esta Deshabilitado momentáneamente"	F
F03-T04	Solicitud de Reserva Incorrecta Cupos Autorizados Insuficiente	1. Ingresar cantidad de cupos en el local. 2. Presionar Botón "Reservar"	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar mensaje de Error: "Local Lleno"	Mostrar mensaje de Error: "Local Lleno"	F
F03-T05	Solicitud de Reserva Incorrecta Cantidad cupos ingresados "Invalido"	1. Ingresar cantidad de cupos en el local. 2. Presionar Botón "Reservar"	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar mensaje de Error: "Ingrese Numero Valido"	Mostrar mensaje de Error: "Ingrese Numero Valido"	F

F-04: Cancelar Reserva (Usuario Persona).**Precondiciones:** - Las reservas deben estar en estado "Pendiente".

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F04-T01	Cancelar Reserva Correcto	1. Presionar botón de reserva a cancelar	Array Reservas Usuario Persona Activo	Reserva pasa a estado "Cancelado" Mostrar solo reservas en estado "Pendiente"	Reserva pasa a estado "Cancelado" Mostrar solo reservas en estado "Pendiente"	P

F-05: Estadística (Usuario Persona)**Precondiciones:** - Tener al menos una reserva "Finalizado"

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F05-T01	Mostrar Estadísticas de Reservas	1. Presionar botón de Estadísticas	Array de Locales Precargados Array de Reservas Usuario Persona Activa	Mostrar en tabla: - Nombre del Local. - Cant. res del Usuario. - Reservas totales del Local. - % de reservas del usuario - Local Favorito	Mostrar en tabla: - Nombre del Local. - Cant. res del Usuario. - Reservas totales del Local. - % de reservas del usuario - Local Favorito	P

F-06: Calificación de local (Usuario Persona)**Precondiciones:** - Reserva en estado "Finalizado".

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F06-T01	Calificar Reserva Correcto	1. Presionar botón de Calificaciones	Array de Reservas Usuario Persona Activa	Mostrar en tabla: - Reservas en estado Finalizado y sin calificación.	Mostrar en tabla: Reservas en estado Finalizado y sin calificación.	P

F-07: Deshabilitar Local (Usuario Local)**Precondiciones:** - Tener al menos una reserva "Finalizada"

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F07-T01	Deshabilitar Local Correcto	1. Seleccionar estado del local 2. Presionar botón "Cambiar"	Array de Locales Precargados Array de Reservas Usuario Local Activa	Cambiar estado del local Mostrar texto estado actual del local	Cambiar estado del local Mostrar texto estado actual del local	P
F07-T02	Deshabilitar Local Incorrecto Local con reservas pendientes	1. Seleccionar estado del local 2. Presionar botón "Cambiar"	Array de Locales Precargados Array de Reservas Usuario Local Activa	NO cambiar estado del local Mostrar texto de error: "El local tiene reservas pendientes"	NO cambiar estado del local Mostrar texto de error: "El local tiene reservas pendientes"	F

F-08: Finalizar Reservas (Usuario Local)**Precondiciones:** - Tener al menos una reserva en estado "Pendiente"

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F08-T01	Finalizar Reserva Correcto	1. Presionar Botón "Mostrar Todas" 2. Presionar botón "Finalizar Reserva"	Array de Locales Precargados Array de Reservas Usuario Local Activa	Cambiar estado la Reserva a "Finalizada"	Cambiar estado la Reserva a "Finalizada"	P

F-09: Cambiar Cupo Máximo del Local (Usuario Local)**Precondiciones:** - Tener al menos una reserva "Finalizada"

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F09-T01	Cambiar Cupo Máximo Correcto	1. Ingresar Nuevo cupo Máximo mayor a 0 2. Presionar Botón "Cambiar"	Array de Locales Precargados Array de Reservas Usuario Local Activa	Cambiar Cupo Máximo del Local	Cambiar Cupo Máximo del Local	P
F09-T02	Cambiar Cupo Máximo Incorrecto Reservas Pendientes	1. Ingresar Nuevo cupo Máximo mayor a 0 2. Presionar Botón "Cambiar"	Array de Locales Precargados Array de Reservas Usuario Local Activa	No Cambiar Cupo Máximo del Local. Mostrar Mensaje de error: "EL Local tiene reservas pendientes"	No Cambiar Cupo Máximo del Local. Mostrar Mensaje de error: "EL Local tiene reservas pendientes"	F
F09-T03	Cambiar Cupo Máximo Incorrecto Nuevo cupo Menor o Igual a 0	1. Ingresar Nuevo cupo Máximo mayor a 0 2. Presionar Botón "Cambiar"	Array de Locales Precargados Array de Reservas Usuario Local Activa	No Cambiar Cupo Máximo del Local. Mostrar Mensaje de error: "Ingrese cupo Mayor a 0"	No Cambiar Cupo Máximo del Local. Mostrar Mensaje de error: "Ingrese cupo Mayor a 0"	F

F-10: Estadística (Usuario Local)**Precondiciones:** - Tener al menos una reserva "Finalizado"

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F10-T01	Mostrar Estadísticas de Reservas y Calificación Correcto	1. Presionar botón de Estadísticas	Array de Locales Precargados Array de Reservas Usuario Local Activa	Mostrar en tabla: - Cupo Máximo - % ocupaciones - Reservas Pendientes - Reservas Finalizadas - Promedio de calificaciones del local	Mostrar en tabla: - Cupo Máximo - % ocupaciones - Reservas Pendientes - Reservas Finalizadas - Promedio de calificaciones del local	P

4. Código HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta    name="viewport"    content="width=device-width,
initial-scale=1.0">

    <title>Obligatorio</title>

    <link rel="stylesheet" href="css/main.css" type="text/css">

</head>

<body>

    <!--Inicio Sesion Persona-->

    <div id="pantallaLoginPersona" class="este">

        <center>

            <h1>Persona</h1>

            <label>Usuario</label>

                <input type="text" id="txtUsuarioPersonaIngresar"
placeholder="Nombre Usuario" class="cajaTextoLogin">

            <br>

            <br>

            <label>Contraseña</label>
```

```

        <input type="password" id="txtPassPersonaIngresar"
placeholder="Contraseña">

        <br>

        <br>

        <p id="etqMensajeErrorInicioPersona"></p>

        <button id="btnIngresarPersona">Ingresar</button>

        <button id="btnRegistroPersona">Registrarme</button>

        <br>

        <br>

        <button id="btnLoginComoLocal">Ingresar como
Local</button>

    </center>

</div>

<!--Inicio Sesion Local-->

<div id="pantallaLoginLocal">

    <center>

        <h1>Local</h1>

        <label>Usuario</label>

        <input type="text" id="txtUsuarioLocalIngresar"
placeholder="Nombre Usuario">

        <br>

        <br>

```

```

        <label>Contraseña</label>

        <input type="password" id="txtPassLocalIngresar"
placeholder="Contraseña">

        <br>

        <p id="etqMensajeErrorInicioLocal"></p>

        <button id="btnIngresarLocal">Ingresar</button>

        <br>

        <br>

        <button id="btnVolverLoginPersona">Ingresar como
Persona</button>

    </center>

</div>

<!--Resgitro Usuario Persona-->

<div id="pantallaRegistroPersona">

    <center>

        <h1>Registro</h1>

        <label>Nombre Completo</label>

        <input type="text" id="txtNombreRegistroPersona"
placeholder="nombre completo">

        <br>

        <br>

        <label>Nombre Usuario</label>

        <input type="text" id="txtUsuarioRegistroPersona"
placeholder="usuario">

        <br>

```



```

        <br>

        <label>Contraseña</label>

        <input type="text" id="txtPassRegistroPersona"
placeholder="ingrese una contraseña">

        <p id="etqMensajeErrorRegistroPersona"></p>

        <p>Condiciones de contraseña:</p>

        <ul>

            <li>1. Tener al menos 6 Caracteres</li>

            <li>2. Tener al menos una mayuscula<br>
                y una minuscula</li>

            <li>3. Contener al menos un numero</li>

        </ul>

        <button
id="btnCargarRegistroPersona">REGISTRARME</button>

        <button id="btnVolverRegistroALogPersona">volver al
login</button>

    </center>

</div>

<!--Pantalla Principal Usuario-->

<div id="pantallaPrincipalPersona">

    <center>

        <h1>Pantalla Prinicpal de Persona</h1>

        <button id="btnVolverPrinPersonaALogin">volver
login</button>

```

```

</center>

<!--Botones opciones Principal Usuario-->

<div id="botonesOpcionesPrincipalPersona">

<button
id="btnPantallaReservasPersona">Reservas</button>

<button
id="btnPantallaPendientesPersona">Pendientes</button>

<button
id="btnPantallaEstadisticasPersona">Estadisticas</button>

<button
id="btnPantallaCalificacionesPersona">Calificaciones</button>

</div>

<!--Pantalla RESERVAS Principal Usuario-->

<div id="pantallaReservasPrincipalPersona">

<fieldset>

<h1>Reservaciones</h1>

<div class="contenedor">

<div class="caja">

<h3>Teatro Solis</h3>

</div>

<div class="caja">



</div>

```

```

        <div class="caja">

            <p>Hacer Reservas</p>

        </div>

        <div class="caja">

            <input type="number"
id="txtCantReservasSolis" placeholder="Cant. Reservas">

            <button
id="btnReservarTestroSolis">Reservar</button><br>

            <a id="etqMensajeErrorReservaSolis"></a>

        </div>

    </div>

    <div class="contenedor">

        <div class="caja">

            <h3>Museo del Carnaval</h3>

        </div>

        <div class="caja">

        </div>

        <div class="caja">

            <p>Hacer Reservas</p>

        </div>

        <div class="caja">

            <input type="number"
id="txtCantReservasMuseoDelCarnaval" placeholder="Cant. Reservas">

```

```

</button>
id="btnReservarMuseoDelCarnaval">Reservar</button><br>

<a
id="etqMensajeErrorReservaMuseoDelCarnaval"></a>

</div>

</div>

<div class="contenedor">

    <div class="caja">

        <h3>Teatro Opera</h3>

    </div>

    <div class="caja">

    </div>

    <div class="caja">

        <p>Hacer Reservas</p>

    </div>

    <div class="caja">

        <input type="number"
id="txtCantReservasTeatroOpera" placeholder="Cant. Reservas">

        </div>

    <div class="caja">

        <button
id="btnReservarTeatroOpera">Reservar</button><br>

        <a
id="etqMensajeErrorReservaTeatroOpera"></a>

    </div>

</div>

```

```

<div class="contenedor">

    <div class="caja">

        <h3>Restaurant Tu Refuerso</h3>

    </div>

    <div class="caja">

    </div>

    <div class="caja">

        <p>Hacer Reservas</p>

    </div>

    <div class="caja">

        <input type="number"
id="txtCantReservasTuRefuerso" placeholder="Cant. Reservas">

        <button
id="btnReservarTuRefuerso">Reservar</button><br>

        <a
id="etqMensajeErrorReservaTuRefuerso"></a>

    </div>

</div>

<div class="contenedor">

    <div class="caja">

        <h3>Restaurant El Galpon</h3>

    </div>

```

```

        <div class="caja">

        </div>

        <div class="caja">

            <p>Hacer Reservas</p>

        </div>

        <div class="caja">

            <input type="number"
id="txtCantReservasElGalpon" placeholder="Cant. Reservas">

            <button
id="btnReservaElGalpon">Reservar</button><br>

            <a id="etqMensajeErrorReservaElGalpon"></a>

        </div>

    </div>

    <div class="contenedor">

        <div class="caja">

            <h3>Museo La Luz</h3>

        </div>

        <div class="caja">

        </div>

        <div class="caja">

            <p>Hacer Reservas</p>

        </div>

```

```

        <div class="caja">

                                <input type="number"
id="txtCantReservasLaLuz" placeholder="Cant. Reservas">

                                <button
id="btnReservarMuseoLaLuz">Reservar</button><br>

                                <a id="etqMensajeErrorReservaLaLuz"></a>

        </div>

</div>

<div class="contenedor">

    <div class="caja">

        <h3>Cafeteria Cafe</h3>

    </div>

    <div class="caja">

    </div>

    <div class="caja">

        <p>Hacer Reservas</p>

    </div>

    <div class="caja">

                                <input type="number"
id="txtCantReservasCafeteriaCafe" placeholder="Cant. Reservas">

                                <button
id="btnReservaCafeteriaCafe">Reservar</button><br>

                                <a
id="etqMensajeErrorReservaCafeteriaCafe"></a>

```

```

        </div>

    </div>

</fieldset>

</div>

<!--Pantalla PENDIENTES Principal Usuario-->

<div id="pantallaPendientesPrincipalPersona">

    <fieldset>

        <h1>Reservas Pendientes</h1>

        <p id="tablaRservasPendientesUsuario">esta es la
tabla</p>

    </fieldset>

</div>

<!--Pantalla ESTADISTICAS Principal Usuario-->

<div id="pantallaEstadisticasPrincipalPersona">

    <fieldset>

        <h1>Pantalla ESTADISTICAS Usuario Persona</h1>

        <p id="tablaEstadisticasUsuario"></p>

        <p id="etqLocalFavorito"></p>

```



```

        </fieldset>

    </div>

    <!--Pantalla CALIFICACIONES Principal Usuario-->

    <div id="pantallaCalificacionesPrincipalPersona">

        <fieldset>

            <h1>Pantalla Reservas Usuario Persona</h1>

            <p id="tablaCalificarReservasUsuario"></p>

        </fieldset>

    </div>

</div>

<!--Pantalla Principal Local-->

<div id="pantallaPrincipalLocal">

    <center>

        <h1>Pantalla Principal Local</h1>

        <button id="btnVolverPrinLocalALogin">Volver
login</button>

    </center>

    <!--Botones opciones Principal Local-->

    <div id="botonesOpcionesPrincipalLocal">

```

```

        <button id="btnPantallaInfoLocal">Informacion
Local</button>

        <button
id="btnPantallaEstadisticasLocal">Estadisticas</button>

        <button id="btnPantallaGestionReservasLocal">Gestion
Reservas</button>

    </div>

    <!--Pantalla INFORMACION LOCAL Principal local-->

    <div id="pantallaPrincipalInfoLocal">

        <fieldset>

            <h1>Pantalla INFO LOCAL Usuario Local</h1>

            <div class="contenedor2">

                <div class="caja">

                    <h3 id="etqNombreLocalInfoLocal"></h3>

                </div>

                <div class="caja">

                    <p>Estado del local:</p>

                    <select
id="txtOpcionesModificacionEstadoLocal">

                        <option>Habilitado</option>

                        <option>Deshabilitado</option>

                    </select>

                    <button
id="btnCambiarEstadoLocal">Cambiar</button>

```

```

                <p id="etqMensajeErrorCambioEstado">Mensaje
Cambio Estado</p>

            </div>

            <div class="caja">

                <p>Estado Actual:<a
id="etqEstadoActualLocal"></a></p>

            </div>

            <div class="caja">

                <p>Ingrese Nuevo Cupo Maximo</p>

                <input type="number"
id="txtModificarCupoMacimoLocal" placeholder="Nuevo Cupo Maximo">

                <button
id="btnModificarCupoMaximoLocal">Cambiar</button><br>

                <a
id="etqMensajeErrorCambioCupoMaxLocal"></a>

                <p>Cupo Max Actual:<a
id="etqCupoMaximoActualLocal"></a></p>

            </div>

        </div>

    </div>

</fieldset>

</div>

<!--Pantalla ESTADISTICAS Principal Local-->

<div id="pantallaPrincipalEstadisticasLocal">

    <fieldset>

        <h1>Pantalla ESTADISTICAS Usuario Local</h1>

```

```

<center>

<table>

    <tr>

        <td>% de ocupacion</td>

        <td>Promedio Calificaciones</td>

        <td>Total Reservas</td>

    </tr>

    <tr>

        <td>

            <p>%<a
id="etqPorcentajeOcupacionLocal"></a></p>

        </td>

        <td>

            <p><a
id="etqPromedioCalificacionesLocal"></a></p>

        </td>

        <td></td>

    </tr>

    <tr>

        <td>

            <p>Cupo Maximo:<a
id="etqCupoMaximoLocal"></a></p>

        </td>

        <td></td>

        <td>

    
```

```

                                <p>Pendientes: <a
id="etqReservasPendientesLocal"></a></p>

                                </td>

                                </tr>

                                <tr>

                                <td>

                                <p>Actuales: <a
id="etqReservasActuales"></a></p>

                                </td>

                                <td></td>

                                <td>

                                <p>Finalizadas <a
id="etqReservasFinalizadasLocal"></a></p>

                                </td>

                                </tr>

                                </table>

                                </center>

                                </fieldset>

                                </div>

                                <!--Pantalla GESTION RESERVAS Principal Local-->

                                <div id="pantallaPrincipalGestionReservasLocal">

                                <fieldset>

                                <h1>Pantalla GESTION RESERVAS Usuario Local</h1>

                                <br>

```

```

        <div class="buscadorReservas">

                                <input type="text"
id="txtTextoBuscadoReservaLocal" placeholder="Buscar Pendientes">

                                <button
id="btnBuscarReservasPendientesLocal">Buscar</button>

        </div>

        <br>

        <div>

                                <button
id="btnMostrarReservasPendientesLocal">Mostrar Todas</button>

        <br>

                                <p id="tblTablaReservasPendientesLocal">la
tabla es esta</p>

        </div>

    </fieldset>

</div>

</div>

<script src="js/index.js"></script>

</body>

</html>

```

5. Código JAVASCRIPT

```
window.addEventListener("load", inicio);

window.addEventListener("load", botonesGeneral);

function inicio() {

    mostrarLoginPersona()

}

function botonesGeneral() {

    asignarBotones();

    asignarBotonesMenu();

}

function ocultarTodo() {

    //Login Persona, Local y registro:

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

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

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

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

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

```

//Ocultar en Pantalla Principal Persona:

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

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

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

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

//Ocultar en Pantalla Principal Local:

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

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

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

function asignarBotones() {

//document.querySelector("#btnIngresarPersona").addEventListener("click", mostrarPrincipalPersona);

//document.querySelector("#btnIngresarLocal").addEventListener("click", mostrarPrincipalLocal);

```



```

document.querySelector("#btnVolverLoginPersona").addEventListener("
click", mostrarLoginPersona);

document.querySelector("#btnVolverRegistroALogPersona").addEventLis
tener("click", mostrarLoginPersona);

document.querySelector("#btnVolverPrinPersonaALogin").addEventListe
ner("click", mostrarLoginPersona);

document.querySelector("#btnLoginComoLocal").addEventListener("clik
k", mostrarLoginLocal);

document.querySelector("#btnVolverPrinLocalALogin").addEventListene
r("click", mostrarLoginLocal);

document.querySelector("#btnRegistroPersona").addEventListener("cli
ck", mostrarRegistroPersona);
}

function asignarBotonesMenu() {

    //Menu Pantalla PERSONA

document.querySelector("#btnPantallaReservasPersona").addEventListe
ner("click", mostrarPantallaReservasPersona);

document.querySelector("#btnPantallaPendientesPersona").addEventLis
tener("click", mostrarPantallaPendientesPersona);

document.querySelector("#btnPantallaEstadisticasPersona").addEventL
istener("click", mostrarPantallaEstadisticasPersona);

```

```

document.querySelector("#btnPantallaCalificacionesPersona").addEventListener("click", mostrarPantallaCalificacionesPersona);

    //menu Pantalla LOCAL

document.querySelector("#btnPantallaInfoLocal").addEventListener("click", mostrarPantallaInfoLocal);

document.querySelector("#btnPantallaEstadisticasLocal").addEventListener("click", mostrarPantallaEstadisticasLocal);

document.querySelector("#btnPantallaGestionReservasLocal").addEventListener("click", mostrarPantallaGestionReservasLocal);
}

function mostrarPrincipalPersona() {

    ocultarTodo();

document.querySelector("#pantallaPrincipalPersona").style.display =
"block";

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

function mostrarLoginPersona() {

    ocultarTodo();

    document.querySelector("#pantallaLoginPersona").style.display =
"block";

```

```

}

function mostrarLoginLocal() {

    ocultarTodo();

    document.querySelector("#pantallaLoginLocal").style.display =
"block";

}

function mostrarRegistroPersona() {

    ocultarTodo();

document.querySelector("#pantallaRegistroPersona").style.display  =
"block";

}

function mostrarPrincipalLocal() {

    ocultarTodo();

    document.querySelector("#pantallaPrincipalLocal").style.display
= "block";

document.querySelector("#botonesOpcionesPrincipalLocal").style.disp
lay = "block";

}

//funciones botones menu Pantalla Principal PERSONA

function mostrarPantallaReservasPersona() {

```

```

        //muestra todo menos div interiores

        ocultarTodo();

document.querySelector("#pantallaPrincipalPersona").style.display =
"block";

document.querySelector("#botonesOpcionesPrincipalPersona").style.di
splay = "block";

        //muestra sub pantalla interior

document.querySelector("#pantallaReservasPrincipalPersona").style.d
isplay = "block";
}

function mostrarPantallaPendientesPersona() {

    ocultarTodo();

document.querySelector("#pantallaPrincipalPersona").style.display =
"block";

document.querySelector("#botonesOpcionesPrincipalPersona").style.di
splay = "block";

document.querySelector("#pantallaPendientesPrincipalPersona").style
.display = "block";

    mostrarReservasPendientesUsuario()
}

function mostrarPantallaEstadisticasPersona() {

```

```

        ocultarTodo();

document.querySelector("#pantallaPrincipalPersona").style.display =
"block";

document.querySelector("#botonesOpcionesPrincipalPersona").style.di
splay = "block";

document.querySelector("#pantallaEstadisticasPrincipalPersona").sty
le.display = "block";
}

function mostrarPantallaCalificacionesPersona() {

    ocultarTodo();

document.querySelector("#pantallaPrincipalPersona").style.display =
"block";

document.querySelector("#botonesOpcionesPrincipalPersona").style.di
splay = "block";

document.querySelector("#pantallaCalificacionesPrincipalPersona").s
tyle.display = "block";

    calificarResvasFinalizadasUsuario()
}

// Funciones Menu Pantalla Principal LOCAL

function mostrarPantallaInfoLocal() {

```

```

        //muestra todo menos div interiores

        ocultarTodo();

        document.querySelector("#pantallaPrincipalLocal").style.display
= "block";

document.querySelector("#botonesOpcionesPrincipalLocal").style.display
= "block";

        //muestra sub pantalla interior

document.querySelector("#pantallaPrincipalInfoLocal").style.display
= "block";

        mostrarDetalleReservaUsuarioLocal()
    }

function mostrarPantallaEstadisticasLocal() {

    ocultarTodo();

    document.querySelector("#pantallaPrincipalLocal").style.display
= "block";

document.querySelector("#botonesOpcionesPrincipalLocal").style.display
= "block";

document.querySelector("#pantallaPrincipalEstadisticasLocal").style
.display = "block";

    mostrarEstadisticasLocal()
}

function mostrarPantallaGestionReservasLocal() {

```

```

        ocultarTodo();

        document.querySelector("#pantallaPrincipalLocal").style.display
= "block";

document.querySelector("#pantallaPrincipalGestionReservasLocal").st
yle.display = "block";
}

//Funcion Registro Persona

class Persona {

    constructor(unId, unNombre, unUsuario, unPass) {

        this.id = unId;

        this.nombre = unNombre;

        this.usuario = unUsuario;

        this.pass = unPass

    }

}

document.querySelector("#btnCargarRegistroPersona").addEventListener
r("click", registrarPersona);

// genero el array

let arrayPersonas = new Array();

```

```

personaTest = new Persona(1, "Nicolas", "nico", "Nicolas123")

personaTest2 = new Persona(2, "Marcos", "marcos", "Marcos123")

personaTest3 = new Persona(3, "Facundo", "facundo", "Facundo123")

personaTest4 = new Persona(4, "Lucia", "lucia", "Lucia123")

personaTest5 = new Persona(5, "Maria", "maria", "Maria123")

personaTest6 = new Persona(6, "Juan", "juan", "Juan123")

personaTest7 = new Persona(7, "Pedro", "pedro", "Pedro123")


arrayPersonas.push(personaTest,      personaTest2,      personaTest3,
personaTest4, personaTest5, personaTest6, personaTest7)


// esto es para que el contador sea automático

let contadorDePersonas = 8;


function repetido(usuario) {

    for (i = 0; i < arrayPersonas.length; i++) {

        if (usuario.toLowerCase() == "") {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "Ingrese Nombre Usuario Valido"

```



```

        return true
    }

    if ((arrayPersonas[i].usuario).toLowerCase() ==
usuario.toLowerCase()) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "Usuario repetido"

        return true
    }

}

for (i = 0; i < arrayLocalesPrecargados.length; i++) {

    if ((arrayLocalesPrecargados[i].usuario).toLowerCase() ==
usuario.toLowerCase()) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "Usuario repetido"

        return true
    }

}

return false
}

function passValido(pass) {

```

```

    if (pass.length < 6) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "El pass debe tener 6 caracteres como minimo";

        return false

    }

    let validarNumerosEnPass = 0

    let validarLetrasMayusculas = 0

    let validarMinusculasEnPass = 0

    for (let i = 0; i < pass.length; i++) {

        // validar si tiene mayuscula

        if (pass.charAt(i) == pass.charAt(i).toUpperCase()) {

            //validar si pass tiene numero

            if (pass.charAt(i) == "0" || pass.charAt(i) == "1" ||
pass.charAt(i) == "2" || pass.charAt(i) == "3" || pass.charAt(i) ==
"4"

                || pass.charAt(i) == "5" || pass.charAt(i) == "6"
|| pass.charAt(i) == "7" || pass.charAt(i) == "8" || pass.charAt(i)
== "9") {

                validarNumerosEnPass++

            }

            else {

                validarLetrasMayusculas++

            }

        }

    }

```

```

        // Validar si pass tiene minuscula

        if (pass.charAt(i) == pass.charAt(i).toLowerCase()) {

            if (pass.charAt(i) == "0" || pass.charAt(i) == "1" ||
pass.charAt(i) == "2" || pass.charAt(i) == "3" || pass.charAt(i) ==
"4"

                || pass.charAt(i) == "5" || pass.charAt(i) == "6"
|| pass.charAt(i) == "7" || pass.charAt(i) == "8" || pass.charAt(i)
== "9") {

                validarNumerosEnPass++

            }

            else {

                validarMinusculasEnPass++

            }

        }

    }

    if (validarLetrasMayusculas == 0) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "El pass debe tener al menos una Mayuscula";

        return false

    }

    if (validarNumerosEnPass == 0) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "El pass debe tener al menos un Numero";

```

```

        return false
    }

    if (validarMinusculasEnPass == 0) {

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "El pass debe tener al menos una Minuscula";

        return false
    }

document.querySelector("#etqMensajeErrorRegistroPersona").innerHTML
= "Felicidades te registraste correctamente!, vuelve a login para
ingresar."

    return true
}

function registrarPersona() {

    // levanto los datos del formulario

        let nombre =
document.querySelector("#txtNombreRegistroPersona").value;

        let usuario =
document.querySelector("#txtUsuarioRegistroPersona").value;

    usuario = usuario.toLowerCase()

```

```

        let pass =
document.querySelector("#txtPassRegistroPersona").value;

        // si el usuario no está repetido creo el objeto y lo agrego

        if (!repetido(usuario) && passValido(pass)) {

            // creo el objeto ... ojo del formulario saco 3 datos, el
otro era el id que es el primero

            let unaPersona = new Persona(contadorDePersonas, nombre,
usuario, pass);

            // agrego el objeto al array

            arrayPersonas.push(unaPersona)

            //incremento el contador de personas

            contadorDePersonas++;

        }

        // reviso a ver si hice las cosas bien

        console.log(arrayPersonas)
    }

    // Dejar ingresados los locales

class Local {

```

```

        constructor(unNombre, unUsuario, unPass, unTipoLocal,
unCupomax, unaFoto, unEstado, unMaximoTotal) {

    this.id = unId;

    this.nombre = unNombre;

    this.usuario = unUsuario;

    this.pass = unPass;

    this.tipo = unTipoLocal;

    this.cupomax = unCupomax;

    this.foto = unaFoto

    this.estado = unEstado

    this.maximoTotal = unMaximoTotal

    unId++

}

}

//falta cargar las imagenes

let arrayLocalesPrecargados = new Array();

let unId = 1;

local1 = new Local("Teatro Solis", "Nicolas123", "Nicolas123",
"teatro", 50, "imgs/foto1.jpg", "habilitado", 100)

local2 = new Local("Museo del Carnaval", "Juan123", "Juan123",
"museo", 50, "imgs/foto2.jpg", "habilitado", 100)

local3 = new Local("Teatro Opera", "Pedro123", "Pedro123",
"teatro", 50, "imgs/foto3.jpg", "habilitado", 100)

local4 = new Local("Tu Refuerzo", "Santiago123", "Santiago123",
"restaurant", 50, "imgs/foto4.jpg", "habilitado", 100)

```

```

local5 = new Local("El Galpon", "Agustin123", "Agustin123",
"restaurant", 50, "imgs/foto5.jpg", "habilitado", 100)

local6 = new Local("Cafeteria Cafe", "Sebastian123",
"Sebastian123", "restaurant", 50, "imgs/foto6.jpg", "habilitado",
100)

local7 = new Local("La Luz", "Ezequiel123", "Ezequiel123", "museo",
50, "imgs/foto7.jpg", "habilitado", 100)

arrayLocalesPrecargados.push(local1, local2, local3, local4,
local5, local6, local7);

console.log(arrayLocalesPrecargados);

//Login Persona

document.querySelector("#btnIngresarPersona").addEventListener("cli
ck", ingresarPersona);

function ingresarPersona() {

    // levanto los datos del formulario

    let usuarioPersonaIngresar =
document.querySelector("#txtUsuarioPersonaIngresar").value;

    let passPersonaIngresar =
document.querySelector("#txtPassPersonaIngresar").value;

    // si el usuario

```

```

document.querySelector("#txtUsuarioPersonaIngresar").value="";

document.querySelector("#txtPassPersonaIngresar").value="";

document.querySelector("#txtCantReservasSolis").value="";

document.querySelector("#txtCantReservasMuseoDelCarnaval").value="";
;

document.querySelector("#txtCantReservasTeatroOpera").value="";

document.querySelector("#txtCantReservasTuRefuerso").value="";

document.querySelector("#txtCantReservasElGalpon").value="";

document.querySelector("#txtCantReservasLaLuz").value="";

document.querySelector("#txtNombreRegistroPersona").value="";

document.querySelector("#txtUsuarioRegistroPersona").value="";

document.querySelector("#txtCantReservasCafeteriaCafe").value="";

document.querySelector("#txtPassRegistroPersona").value="";


        if    (!usuarioDePersonaRegistrado(usuarioPersonaIngresar,
arrayPersonas)    &&    !passCorrectaDePersona(passPersonaIngresar,
arrayPersonas)) {

            //que te deje entrar si se cumple

                                usuarioPersonaActivo    =
usuarioPersonaIngresar.toLowerCase();

            mostrarPrincipalPersona()

        }

}

function usuarioDePersonaRegistrado(usuario, array) {

```



```

        for (i = 0; i < array.length; i++) {

            if ((array[i].usuario).toLowerCase() ==
usuario.toLowerCase()) {

                return false

            }

        }

document.querySelector("#etqMensajeErrorInicioPersona").innerHTML =
"Nombre de usuario invalido o inexistente"

        document.querySelector("#etqMensajeErrorInicioLocal").innerHTML
= "Nombre de usuario invalido o inexistente"

        return true
    }

function passCorrectaDePersona(pass, array) {

    for (i = 0; i < array.length; i++) {

        if (pass == "") {

            return true

        }

        if (array[i].pass == pass) {

            return false

        }

    }

}

```

```

document.querySelector("#etqMensajeErrorInicioPersona").innerHTML =
"Contraseña Invalida"

    document.querySelector("#etqMensajeErrorInicioLocal").innerHTML
= "Contraseña Invalida"

    return true
}

// Login Local

let usuarioLocalActivo = ""

let passLocalActivo = ""

document.querySelector("#btnIngresarLocal").addEventListener("click
", ingresarLocal);

function ingresarLocal() {

    // levanto los datos del formulario

        let          usuarioLocalIngresar          =
document.querySelector("#txtUsuarioLocalIngresar").value;

        let          passLocalIngresar          =
document.querySelector("#txtPassLocalIngresar").value;

        document.querySelector("#txtUsuarioLocalIngresar").value="";

        document.querySelector("#txtPassLocalIngresar").value="";

document.querySelector("#txtModificarCupoMacimoLocal").value="";

```

```

document.querySelector("#txtTextoBuscadoReservaLocal").value="";

    // si el usuario

        if (!usuarioDePersonaRegistrado(usuarioLocalIngresar,
arrayLocalesPrecargados)                                &&
!passCorrectaDePersona(passLocalIngresar, arrayLocalesPrecargados))
{

    //que te deje entrar si se cumple

    uruarioLocalIng = usuarioLocalIngresar;

    usuarioLocalActivo = usuarioLocalIngresar

    passLocalActivo = passLocalIngresar

    mostrarPrincipalLocal()

}

}

// Objeto Reservas

class Reserva {

    constructor(unLocal, unUsuario, unCantidadRes, unEstado,
unNombreUsuarioLocal, unCalificacion) {

        this.local = unLocal

        this.usuario = unUsuario

        this.cantidadRes = unCantidadRes

        this.estado = unEstado

        this.nombreUsuarioLocal = unNombreUsuarioLocal

```

```

        this.calificacion = unCalificacion
    }
}

let arrayReservas = new Array();

let reserva1 = new Reserva("Teatro Solis", "nico", 10, "Pendiente",
"nicolas123");

let reserva2 = new Reserva("Museo del Carnaval", "nico", 10,
"Pendiente", "juan123");

let reserva3 = new Reserva("Teatro Opera", "nico", 10, "Pendiente",
"pedro123");

let reserva4 = new Reserva("Tu Refuerso", "nico", 10, "Pendiente",
"santiago123");

let reserva5 = new Reserva("El Galpon", "nico", 10, "Pendiente",
"agustin123");

let reserva6 = new Reserva("Cafeteria Cafe", "nico", 10,
"Pendiente", "sebastian123");

let reserva7 = new Reserva("La Luz", "nico", 10, "Pendiente",
"ezequiel123");

let reserva8 = new Reserva("Teatro Solis", "marcos", 10,
"Pendiente", "nicolas123");

let reserva9 = new Reserva("Museo del Carnaval", "marcos", 10,
"Pendiente", "juan123");

let reserva10 = new Reserva("Teatro Opera", "marcos", 10,
"Pendiente", "pedro123");

```

```

let reserva11= new Reserva("Tu Refuerso", "marcos", 10,
"Pendiente", "santiago123");

let reserva12= new Reserva("El Galpon", "marcos", 10, "Pendiente",
"agustin123");

let reserva13= new Reserva("Cafeteria Cafe", "marcos", 10,
"Pendiente", "sebastian123");

let reserva14= new Reserva("La Luz", "marcos", 10, "Pendiente",
"ezequiel123");


let reserva15 = new Reserva("Teatro Solis", "facundo", 10,
"Pendiente", "nicolas123");

let reserva16 = new Reserva("Museo del Carnaval", "facundo", 10,
"Pendiente", "juan123");

let reserva17 = new Reserva("Teatro Opera", "facundo", 10,
"Pendiente", "pedro123");

let reserva18= new Reserva("Tu Refuerso", "facundo", 10,
"Pendiente", "santiago123");

let reserva19= new Reserva("El Galpon", "facundo", 10, "Pendiente",
"agustin123");

let reserva20= new Reserva("Cafeteria Cafe", "facundo", 10,
"Pendiente", "sebastian123");

let reserva21= new Reserva("La Luz", "facundo", 10, "Pendiente",
"ezequiel123");


let reserva22 = new Reserva("Teatro Solis", "lucia", 10,
"Pendiente", "nicolas123");

let reserva23 = new Reserva("Museo del Carnaval", "lucia", 10,
"Pendiente", "juan123");

let reserva24 = new Reserva("Teatro Opera", "lucia", 10,
"Pendiente", "pedro123");

```

```

let reserva25= new Reserva("Tu Refuerzo", "lucia", 10, "Pendiente",
"santiago123");

let reserva26= new Reserva("El Galpon", "lucia", 10, "Pendiente",
"agustin123");

let reserva27= new Reserva("Cafeteria Cafe", "lucia", 10,
"Pendiente", "sebastian123");

let reserva28= new Reserva("La Luz", "lucia", 10, "Pendiente",
"ezequiel123");


let reserva29 = new Reserva("Teatro Solis", "lucia", 10,
"Finalizada", "nicolas123", 4);

let reserva30 = new Reserva("Museo del Carnaval", "lucia", 10,
"Finalizada", "juan123", 4);

let reserva31 = new Reserva("Teatro Opera", "lucia", 10,
"Finalizada", "pedro123", 4);

let reserva32= new Reserva("Tu Refuerzo", "lucia", 10,
"Finalizada", "santiago123", 4);

let reserva33= new Reserva("El Galpon", "lucia", 10, "Finalizada",
"agustin123", 4);

let reserva34= new Reserva("Cafeteria Cafe", "lucia", 10,
"Finalizada", "sebastian123", 4);

let reserva35= new Reserva("La Luz", "lucia", 10, "Finalizada",
"ezequiel123", 4);


let reserva36 = new Reserva("Teatro Solis", "maria", 10,
"Pendiente", "nicolas123");

let reserva37 = new Reserva("Museo del Carnaval", "maria", 10,
"Pendiente", "juan123");

let reserva38 = new Reserva("Teatro Opera", "maria", 10,
"Pendiente", "pedro123");

```

```

let reserva39= new Reserva("Tu Refuerso", "maria", 10, "Pendiente",
"santiago123");

let reserva40= new Reserva("El Galpon", "maria", 10, "Pendiente",
"agustin123");

let reserva41= new Reserva("Cafeteria Cafe", "maria", 10,
"Pendiente", "sebastian123");

let reserva42= new Reserva("La Luz", "maria", 10, "Pendiente",
"ezequiel123");


let reserva43 = new Reserva("Teatro Solis", "maria", 10,
"Finalizada", "nicolas123", 4);

let reserva44 = new Reserva("Museo del Carnaval", "maria", 10,
"Finalizada", "juan123", 4);

let reserva45 = new Reserva("Teatro Opera", "maria", 10,
"Finalizada", "pedro123", 4);

let reserva46= new Reserva("Tu Refuerso", "maria", 10,
"Finalizada", "santiago123", 4);

let reserva47= new Reserva("El Galpon", "maria", 10, "Finalizada",
"agustin123", 4);

let reserva48= new Reserva("Cafeteria Cafe", "maria", 10,
"Finalizada", "sebastian123", 4);

let reserva49= new Reserva("La Luz", "maria", 10, "Finalizada",
"ezequiel123", 4);


arrayReservas.push(reserva1, reserva2, reserva3, reserva4,
reserva5, reserva6, reserva7, reserva8, reserva9, reserva10,
reserva11, reserva12, reserva13, reserva14, reserva15, reserva16,
reserva17, reserva18, reserva19, reserva20, reserva21, reserva22,
reserva23, reserva24, reserva25, reserva26, reserva27, reserva28,
reserva29, reserva30, reserva31, reserva32, reserva33, reserva34,

```

```

reserva35, reserva36,reserva37, reserva38, reserva39, reserva40,
reserva41, reserva42,reserva43, reserva44, reserva45, reserva46,
reserva47, reserva48,reserva49)

// Validar Reservas Persona.

// Escucha de Botones Reservas

let usuarioPersonaActivo = ""

document.querySelector("#btnReservarTestroSolis").addEventListener(
"click", reservarLugarTeatroSolis);

document.querySelector("#btnReservaCafeteriaCafe").addEventListener(
"click", reservarLugarCafeteriaCafe);

document.querySelector("#btnReservarMuseoDelCarnaval").addEventListener(
"click", reservarLugarMuseoDelCarnaval);

document.querySelector("#btnReservarTeatroOpera").addEventListener(
"click", reservarLugarTeatroOpera);

document.querySelector("#btnReservarTuRefuerso").addEventListener(
"click", reservarLugarTuRefuerso);

document.querySelector("#btnReservarMuseoLaLuz").addEventListener(
"click", reservarLugarMuseoDeLuz);

document.querySelector("#btnReservaElGalpon").addEventListener("cli
ck", reservaLugarElGalpon);

```



```
// Funciones de cada local para hacer las reservas

function reservarLugarTeatroSolis() {

    reservarLugar("Teatro Solis", "#txtCantReservasSolis",
"#etqMensajeErrorReservaSolis", "nicolas123")

}

function reservarLugarMuseoDelCarnaval() {

    reservarLugar("Museo del Carnaval",
"#txtCantReservasMuseoDelCarnaval",
"#etqMensajeErrorReservaMuseoDelCarnaval", "juan123")

}

function reservarLugarTeatroOpera() {

    reservarLugar("Teatro Opera", "#txtCantReservasTeatroOpera",
"#etqMensajeErrorReservaTeatroOpera", "pedro123")

}

function reservarLugarTuRefuerso() {

    reservarLugar("Tu Refuerso", "#txtCantReservasTuRefuerso",
"#etqMensajeErrorReservaTuRefuerso", "santiago123")

}

function reservarLugarMuseoDeLuz() {

    reservarLugar("La Luz", "#txtCantReservasLaLuz",
"#etqMensajeErrorReservaLaLuz", "ezequiel123")

}
```

```

function reservarLugarCafeteriaCafe() {

    reservarLugar("Cafeteria      Cafe",
"#txtCantReservasCafeteriaCafe",
"#etqMensajeErrorReservaCafeteriaCafe", "sebastian123")

}

function reservaLugarElGalpon() {

    reservarLugar("El   Galpon",   "#txtCantReservasElGalpon",
"#etqMensajeErrorReservaElGalpon", "agustin123")

}

// Funcion que genera la reserva

function reservarLugar(nombre, texto, etqError, usuarioDelLocal) {

    let cantidadReservas = 0;

    if    (document.querySelector(texto).value  ==  ""    ||
document.querySelector(texto).value == "0") {

        document.querySelector(etqError).innerHTML = "Ingrese
numero Valido"

    }

    else {

                                cantidadReservas      =
parseInt(document.querySelector(texto).value);

        if    (!validarReservas(cantidadReservas,  nombre,
usuarioPersonaActivo, etqError)) {

```

```

        // creo el objeto

        let reserva = new Reserva(nombre, usuarioPersonaActivo,
cantidadReservas, "Pendiente", usuarioDelLocal);

        // agrego el objeto al array

        arrayReservas.push(reserva)

    }

}

}

// Funcion qpara validar las reservas que se estan cargando, de ser
correcto da False, sino se puede cargar da True.

function validarReservas(cantidadReservas, localDeReserba,
usuarioPersonaActivo, etqError) {

    for (i = 0; i < arrayLocalesPrecargados.length; i++) {

        if (arrayLocalesPrecargados[i].nombre == localDeReserba) {

            if (arrayLocalesPrecargados[i].estado == "habilitado")
{

                if ((arrayLocalesPrecargados[i].cupomax -
cantidadReservas) >= 0) {

                    for (x = 0; x < arrayReservas.length; x++) {

```

```

                                if (arrayReservas[x].usuario ==
usuarioPersonaActivo && arrayReservas[x].local == localDeReserba &&
arrayReservas[x].estado == "Pendiente") {

document.querySelector(etqError).innerHTML = "Ya hay una reserva
Pendiente"

                                return true

                                }

                                }

                                document.querySelector(etqError).innerHTML =
"Reserva Exitosa"

                                arrayLocalesPrecargados[i].cupomax =
arrayLocalesPrecargados[i].cupomax - cantidadReservas

                                return false

                                } else {

                                document.querySelector(etqError).innerHTML =
"Local lleno"

                                return true

                                }

                                } else {

                                document.querySelector(etqError).innerHTML = "El
local está deshabilitado momentáneamente"

                                return true

                                }

```

```

    }

    }

}

// Funcion modifica informacion de local, segun usuarioo que
ingrese que info muestra en pantalla, si un local o otro.

document.querySelector("#btnModificarCupoMaximoLocal").addEventListener("click", modificarCupoMaximoLocal);

document.querySelector("#btnCambiarEstadoLocal").addEventListener("click", modificarEstadoLocal)

let uruarioLocalIng = ""

let posicionDeLocalIngresado = ""

let totalReservasPendientes = 0

function mostrarDetalleReservaUsuarioLocal() {

    for (i = 0; i < arrayLocalesPrecargados.length; i++) {

        if (arrayLocalesPrecargados[i].usuario.toLowerCase() ==
uruarioLocalIng.toLowerCase()) {

document.querySelector("#etqNombreLocalInfoLocal").innerHTML =
arrayLocalesPrecargados[i].nombre;

```

```

        //modificarCupoMaximoLocal (i)

        posicionDeLocalIngresado = i

document.querySelector("#etqEstadoActualLocal").innerHTML =
arrayLocalesPrecargados[posicionDeLocalIngresado].estado

document.querySelector("#etqCupoMaximoActualLocal").innerHTML =
arrayLocalesPrecargados[posicionDeLocalIngresado].cupomax

document.querySelector("#etqMensajeErrorCambioCupoMaxLocal").innerH
TML = ""

document.querySelector("#etqMensajeErrorCambioEstado").innerHTML =
"Mensaje Cambio Estado"

    }

}

}

function modificarEstadoLocal() {

        let estadoActualLocal =
document.querySelector("#txtOpcionesModificacionEstadoLocal").value

        estadoActualLocal = estadoActualLocal.toLowerCase()

        for (let x = 0; x < arrayReservas.length; x++) {

                if (arrayReservas[x].local ==
arrayLocalesPrecargados[posicionDeLocalIngresado].nombre &&
arrayReservas[x].estado == "Pendiente") {

```

```

        totalReservasPendientes = totalReservasPendientes +
arrayReservas[x].cantidadRes

    }

}

if (totalReservasPendientes == 0) {

    if (estadoActualLocal == "habilitado") {

arrayLocalesPrecargados[posicionDeLocalIngresado].estado =
"habilitado"

document.querySelector("#etqEstadoActualLocal").innerHTML =
"Habilitado"

document.querySelector("#etqMensajeErrorCambioEstado").innerHTML =
"Cambio realizado"

    }

    else {

arrayLocalesPrecargados[posicionDeLocalIngresado].estado =
"deshabilitado"

document.querySelector("#etqEstadoActualLocal").innerHTML =
"Deshabilitado"

document.querySelector("#etqMensajeErrorCambioEstado").innerHTML =
"Cambio realizado"

    }

}

```

```

    }

    else {

document.querySelector("#etqMensajeErrorCambioEstado").innerHTML =
"El local tiene reservas pendientes"

        totalReservasPendientes = 0

    }

}

function modificarCupoMaximoLocal() {

                                let                nuevoCupoIng                =
parseInt(document.querySelector("#txtModificarCupoMacimoLocal").value)

    for (let x = 0; x < arrayReservas.length; x++) {

                                if                (arrayReservas[x].local                ==
arrayLocalesPrecargados[posicionDeLocalIngresado].nombre                &&
arrayReservas[x].estado == "Pendiente") {

                                totalReservasPendientes = totalReservasPendientes +
arrayReservas[x].cantidadRes

                                }

    }

    if (totalReservasPendientes == 0 && nuevoCupoIng > 0) {

        arrayLocalesPrecargados[posicionDeLocalIngresado].cupomax =
nuevoCupoIng
    }
}

```



```

arrayLocalesPrecargados[posicionDeLocalIngresado].maximoTotal      =
nuevoCupoIng

document.querySelector("#etqCupoMaximoActualLocal").innerHTML      =
nuevoCupoIng

document.querySelector("#etqMENsajeErrorCambioCupoMaxLocal").innerH
TML = "Cambio de cupos realizado"

    }

    else if (nuevoCupoIng <= 0) {

document.querySelector("#etqMENsajeErrorCambioCupoMaxLocal").innerH
TML = "Ingrese Cupo Mayor a 0"

    }

    else {

document.querySelector("#etqMENsajeErrorCambioCupoMaxLocal").innerH
TML = "El local tiene reservas pendientes"

        totalReservasPendientes = 0

    }

}

// Mostrar Reservas Pendietes y Finalizadas Usuario Local

```

```

document.querySelector("#btnMostrarReservasPendientesLocal").addEventListener("click", mostrarReservasPendientesLocal);

function mostrarReservasPendientesLocal() {

    let tablaPendientes = "<table border=0>"

    tablaPendientes += `<tr>

                                <td>Usuario</td>

                                <td>Nombre Local</td>

                                <td>Lugares Reservados</td>

                                <td>Estado Reserva</td>`

    console.log(arrayReservas)

    for (let x = 0; x < arrayReservas.length; x++) {

        if (arrayReservas[x].estado == "Pendiente" &&
arrayReservas[x].nombreUsuarioLocal.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

            tablaPendientes += `<tr>

<td>${arrayReservas[x].usuario}</td>

                                <td>${arrayReservas[x].local}</td>

<td>${arrayReservas[x].cantidadRes}</td>

```

```

                                <td>${arrayReservas[x].estado}</td>

                                <td><button
id="btnCancelar${arrayReservas[x].usuario}"
name="${arrayReservas.estado}"
onclick="finalizarReservasLocal(${x})">Finalizar
Reserva</button></td>

                                </tr>`

        }

    }

    for (let z = 0; z < arrayReservas.length; z++) {

        if (arrayReservas[z].estado == "Finalizada" &&
arrayReservas[z].nombreUsuarioLocal.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

            tablaPendientes += `<tr>

<td>${arrayReservas[z].usuario}</td>

                                <td>${arrayReservas[z].local}</td>

<td>${arrayReservas[z].cantidadRes}</td>

                                <td>${arrayReservas[z].estado}</td>

                                </tr>`

        }

    }

    tablaPendientes += `</table>`

```

```

document.querySelector("#tblTablaReservasPendientesLocal").innerHTML
L = tablaPendientes

}

// Funcion Cancelar Reservas Local

function finalizarReservasLocal(reservaCancelada) {

    if (arrayReservas[reservaCancelada].estado == "Pendiente") {

        arrayReservas[reservaCancelada].estado = "Finalizada"

        for (let i = 0; i < arrayLocalesPrecargados.length; i++) {

            if (arrayLocalesPrecargados[i].usuario.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

                arrayLocalesPrecargados[i].cupomax =
arrayLocalesPrecargados[i].cupomax
arrayReservas[reservaCancelada].cantidadRes

            }

        }

    }

    mostrarReservasPendientesLocal()

```

```

}

// Mostrar Reservas Pendientes Usuario

function mostrarReservasPendientesUsuario() {

    let tablaPendientes = "<table border=0>"

    tablaPendientes += `<tr>

                        <td>Usuario</td>

                        <td>Nombre Local</td>

                        <td>Lugares Reservados</td>

                        <td>Estado Reserva</td>`

    console.log(arrayReservas)

    for (let x = 0; x < arrayReservas.length; x++) {

        if (arrayReservas[x].estado == "Pendiente" &&
arrayReservas[x].usuario.toLowerCase() ==
usuarioPersonaActivo.toLowerCase()) {

            tablaPendientes += `<tr>

<td>${arrayReservas[x].usuario}</td>

                        <td>${arrayReservas[x].local}</td>

```

```

<td>${arrayReservas[x].cantidadRes}</td>

<td>${arrayReservas[x].estado}</td>

<td><button
id="btnCancelar${arrayReservas[x].usuario}"
name="${arrayReservas.estado}"
onclick="finalizarReservasUsuario(${x})">Cancelar
Reserva</button></td>

</tr>`

    }

}

tablaPendientes += `</table>`

document.querySelector("#tablaReservasPendientesUsuario").innerHTML
= tablaPendientes

}

// Funcion Cancelar Reservas usuario

function finalizarReservasUsuario(reservaCancelada) {

    if (arrayReservas[reservaCancelada].estado == "Pendiente") {

        arrayReservas[reservaCancelada].estado = "Cancelada"

```

```

        for (let i = 0; i < arrayLocalesPrecargados.length; i++) {

            if (arrayLocalesPrecargados[i].nombre ==
arrayReservas[reservaCancelada].local) {

                arrayLocalesPrecargados[i].cupomax =
arrayLocalesPrecargados[i].cupomax
arrayReservas[reservaCancelada].cantidadRes

            }

        }

    }

    mostrarReservasPendientesUsuario()

}

// Calificar Reservas Finalizadas Usuaio

function calificarResvasFinalizadasUsuario() {

    let tablaPendientes = "<table border=0>"

    tablaPendientes += `<tr>

        <td>Usuario</td>

```

```

        <td>Nombre Local</td>

        <td>Lugares Reservados</td>

        <td>Estado Reserva</td>`

console.log(arrayReservas)

for (let x = 0; x < arrayReservas.length; x++) {

    if (arrayReservas[x].estado == "Finalizada" &&
arrayReservas[x].usuario.toLowerCase() ==
usuarioPersonaActivo.toLowerCase() && arrayReservas[x].calificacion
== undefined) {

        tablaPendientes += `<tr>

<td>${arrayReservas[x].usuario}</td>

        <td>${arrayReservas[x].local}</td>

<td>${arrayReservas[x].cantidadRes}</td>

        <td>${arrayReservas[x].estado}</td>

        <td><select
id="txtCalificacionPosibles${x}">

            <option> 1 </option>

            <option> 2 </option>

            <option> 3 </option>

            <option> 4 </option>

            <option> 5 </option>

        </select></td>

```



```

<td><button
id="btnCalificacionPosibles${x}" name="arrayReservas[${x}].usuario"
onclick="calidifcarResevasUsuario (${x}) ">Calificar
Reserva</button></td>

</tr>`

    }

}

tablaPendientes += `</table>`

document.querySelector("#tablaCalificarReservasUsuario").innerHTML
= tablaPendientes

}

// Funcion Calificar Reservas Usuario

function calidifcarResevasUsuario(x) {

    let numIng =
parseInt(document.querySelector(`#txtCalificacionPosibles${x}` ).value)

    arrayReservas[x].calificacion = numIng

```

```

        calificarResvasFinalizadasUsuario()

    }

// Buscador Reservas Pendientes Local

document.querySelector("#btnBuscarReservasPendientesLocal").addEventListener("click", buscarReservasPendientesLocal);

function buscarReservasPendientesLocal() {

    document.querySelector("#tblTablaReservasPendientesLocal").innerHTML = ""

    let bandera=0;

    let textoIng = document.querySelector("#txtTextoBuscadoReservaLocal").value;

    tablaPendientes = "<table border=1>"

```

```

tablaPendientes += `<tr>

                                <td>Usuario</td>

                                <td>Nombre Local</td>

                                <td>Lugares Reservados</td>

                                <td>Estado Reserva</td>`

for (let i = 0; i < arrayReservas.length; i++) {

    if (arrayReservas[i].usuario.indexOf(textoIng) >= 0) {

        if (arrayReservas[i].estado == "Pendiente" &&
arrayReservas[i].nombreUsuarioLocal.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

            tablaPendientes += `<tr>

                                <td>${arrayReservas[i].usuario}</td>

                                <td>${arrayReservas[i].local}</td>

                                <td>${arrayReservas[i].cantidadRes}</td>

                                <td>${arrayReservas[i].estado}</td>

                                <td><button
id="btnCancelar${arrayReservas[i].usuario}"
name="${arrayReservas[i].estado}"
onclick="finalizarReservasLocal(${i})">Finalizar
Reserva</button></td>

                                </tr>`

            bandera++

        }

    }

}

```

```

    }

    for (let i = 0; i < arrayReservas.length; i++) {

        if (arrayReservas[i].usuario.indexOf(textoIng) >= 0) {

            if (arrayReservas[i].estado == "Finalizada" &&
arrayReservas[i].nombreUsuarioLocal.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

                tablaPendientes += `<tr>

                    <td>${arrayReservas[i].usuario}</td>

                    <td>${arrayReservas[i].local}</td>

<td>${arrayReservas[i].cantidadRes}</td>

                    <td>${arrayReservas[i].estado}</td>

                    </tr>`

            }

        }

    }

    tablaPendientes += `</table>`

document.querySelector("#tblTablaReservasPendientesLocal").innerHTM
L = tablaPendientes

if (bandera==0) {

    alert("No hay reservas en estado Pendiente")

```

```

}

}

// Estadísticas Usuario Local

function mostrarEstadísticasLocal() {

    let resPendiente = 0

    let resFinalizadas = 0

    let promCalificacion = 0

    let contador = 0

    let cupoMaximo = 0

    cupomaximo = cupoMaximo + (resPendiente + resFinalizadas)

    let ocupacion = (resPendiente * 100) / cupoMaximo

    document.querySelector("#etqFotoLocalEstadísticas").innerHTML=
    ""

    for (let i = 0; i < arrayReservas.length; i++) {

        if (arrayReservas[i].nombreUsuarioLocal.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

```

```

        if (arrayReservas[i].estado == "Pendiente") {

            resPendiente = resPendiente +
parseInt(arrayReservas[i].cantidadRes)

        }

        if (arrayReservas[i].estado == "Finalizada") {

            resFinalizadas = resFinalizadas +
parseInt(arrayReservas[i].cantidadRes)

        }

        if (arrayReservas[i].calificacion != undefined) {

            contador++

            promCalificacion += arrayReservas[i].calificacion

        }

        // mostrar nombre del local

document.querySelector("#etqNombreLocalEstadisticasMostradas").inne
rHTML+= arrayReservas[i].local

    }

}

// Mostrar foto del local de estadisticas

for (let x=0; x < arrayLocalesPrecargados.length; x++){

```

```

        let image = document.createElement('img')

        image.src = arrayLocalesPrecargados[x].foto

        image.height = 100

        if (arrayLocalesPrecargados[x].usuario.toLowerCase() ==
usuarioLocalActivo.toLowerCase()){

document.querySelector("#etqFotoLocalEstadisticas").appendChild(ima
ge)

        }

    }

    for (let x = 0; x < arrayLocalesPrecargados.length; x++) {

        if (arrayLocalesPrecargados[x].usuario.toLowerCase() ==
usuarioLocalActivo.toLowerCase()) {

                                                    cupoMaximo =
parseInt(arrayLocalesPrecargados[x].maximoTotal)

        }

    }

    // % ocupacion

document.querySelector("#etqPorcentajeOcupacionLocal").innerHTML =
resPendiente * 100 / cupoMaximo

```

```

        // Cupo Maximo

        document.querySelector("#etqCupoMaximoLocal").innerHTML =
cupoMaximo

        // Promedio Calificaion

document.querySelector("#etqPromedioCalificacionesLocal").innerHTML
= (promCalificacion / contador).toFixed(1)

        // Res Pendientes

        document.querySelector("#etqReservasPendientesLocal").innerHTML
= resPendiente

        // Res finalizadas

document.querySelector("#etqReservasFinalizadasLocal").innerHTML =
resFinalizadas

        // Res Actuales/Totales

        document.querySelector("#etqReservasActuales").innerHTML =
resPendiente

    }

//Estadisticas Persona

```



```

document.querySelector("#btnPantallaEstadisticasPersona").addEventListener("click", buscarEstadisticasPersona);

function buscarEstadisticasPersona() {

    document.querySelector("#tablaEstadisticasUsuario").innerHTML =
    ""

    tablaEstadisticasPersona = "<table border=1>"

    tablaEstadisticasPersona += `<tr>

                                <td>Locales</td>

                                <td>Cantidad</td>

                                <td>Total Reservas</td>

                                <td>%Reserva</td>`

    for (let i = 0; i < arrayLocalesPrecargados.length; i++) {

        let cantidadReservasEstadisticasPersona = 0;

        let confirmacion = 0;

        let reservasTotalesLocal = 0;

        for (let x = 0; x < arrayReservas.length; x++) {

```

```

                if (arrayLocalesPrecargados[i].nombre ==
arrayReservas[x].local) {

                    reservasTotalesLocal++

                    if (arrayReservas[x].usuario.toLowerCase() ==
usuarioPersonaActivo.toLowerCase() &&
arrayLocalesPrecargados[i].nombre == arrayReservas[x].local) {

                        cantidadReservasEstadisticasPersona++

                        if (arrayReservas[x].estado == "Finalizada") {

                            confirmacion++

                        }

                    }

                }

            }

        }

        if (confirmacion > 0) {

            tablaEstadisticasPersona += `<tr>

                <td>${arrayLocalesPrecargados[i].nombre}</td>

                <td>${cantidadReservasEstadisticasPersona}</td>

                <td>${reservasTotalesLocal}</td>

```

```

                <td>%${((100 * cantidadReservasEstadisticasPersona)
/ reservasTotalesLocal).toFixed(1)}</td>

            </tr>`

            confirmacion = 0;

            localFavorito()

        }

    }

    tablaEstadisticasPersona += `</table>`

    document.querySelector("#tablaEstadisticasUsuario").innerHTML =
tablaEstadisticasPersona

}

class Favorito {

    constructor(unLocal, unReservas) {

        this.local = unLocal

        this.cantidadReservas = unReservas

```

```

    }

}

let cantidadResFav = new Array()

function localFavorito() {

    document.querySelector("#etqLocalFavorito").innerHTML =
"Locales mas reservados por el usuario: "

    for (let i = 0; i < arrayLocalesPrecargados.length; i++) {

        let bandera = 0;

        for (let x = 0; x < arrayReservas.length; x++) {

            if (arrayReservas[x].usuario.toLowerCase() ==
usuarioPersonaActivo.toLowerCase() &&
arrayLocalesPrecargados[i].nombre == arrayReservas[x].local) {

                personaTest = new
Favorito(arrayLocalesPrecargados[i].nombre, 1)

                for (let z = 0; z < cantidadResFav.length; z++) {

```

```
                if (cantidadResFav[z].local ==  
arrayLocalesPrecargados[i].nombre) {  
  
                    cantidadResFav[z].cantidadReservas++  
  
                    bandera++  
  
                }  
  
            }  
  
            if (bandera == 0) {  
  
                cantidadResFav.push(personaTest)  
  
            }  
  
        }  
  
    }  
  
}  
  
let mayor = 0;  
  
for (let max of cantidadResFav) {  
  
    if (max.cantidadReservas > mayor) {  
  
        mayor = max.cantidadReservas
```

```

    }

}

for (let fav of cantidadResFav) {

    if (fav.cantidadReservas == mayor) {

        document.querySelector("#etqLocalFavorito").innerHTML
+= `<br></br>` + fav.local

    }

    fav.cantidadReservas = 0

}

}

```

6. LISTA INFORMACIÓN PRECARGADA

Usuario Pre-Cargados:

Nombre / Usuario / Contraseña

```
personaTest = new Persona(1, "Nicolas", "nico", "Nicolas123")  
  
personaTest2 = new Persona(2, "Marcos", "marcos", "Marcos123")  
  
personaTest3 = new Persona(3, "Facundo", "facundo", "Facundo123")  
  
personaTest4 = new Persona(4, "Lucia", "lucia", "Lucia123")  
  
personaTest5 = new Persona(5, "Maria", "maria", "Maria123")  
  
personaTest6 = new Persona(6, "Juan", "juan", "Juan123")  
  
personaTest7 = new Persona(7, "Pedro", "pedro", "Pedro123")
```

Reservas Pre-Cargadas:

Local / Usuario / CantidadRes / Estado / NombreUsuarioLocal / Calificacion

```
let arrayReservas = new Array();  
  
let reserva1 = new Reserva("Teatro Solis", "nico", 10, "Pendiente",  
"nicolas123");  
  
let reserva2 = new Reserva("Museo del Carnaval", "nico", 10,  
"Pendiente", "juan123");  
  
let reserva3 = new Reserva("Teatro Opera", "nico", 10, "Pendiente",  
"pedro123");
```

```

let reserva4 = new Reserva("Tu Refuerso", "nico", 10, "Pendiente",
"santiago123");

let reserva5 = new Reserva("El Galpon", "nico", 10, "Pendiente",
"agustin123");

let reserva6 = new Reserva("Cafeteria Cafe", "nico", 10,
"Pendiente", "sebastian123");

let reserva7 = new Reserva("La Luz", "nico", 10, "Pendiente",
"ezequiel123");


let reserva8 = new Reserva("Teatro Solis", "marcos", 10,
"Pendiente", "nicolas123");

let reserva9 = new Reserva("Museo del Carnaval", "marcos", 10,
"Pendiente", "juan123");

let reserva10 = new Reserva("Teatro Opera", "marcos", 10,
"Pendiente", "pedro123");

let reserva11= new Reserva("Tu Refuerso", "marcos", 10,
"Pendiente", "santiago123");

let reserva12= new Reserva("El Galpon", "marcos", 10, "Pendiente",
"agustin123");

let reserva13= new Reserva("Cafeteria Cafe", "marcos", 10,
"Pendiente", "sebastian123");

let reserva14= new Reserva("La Luz", "marcos", 10, "Pendiente",
"ezequiel123");


let reserva15 = new Reserva("Teatro Solis", "facundo", 10,
"Pendiente", "nicolas123");

let reserva16 = new Reserva("Museo del Carnaval", "facundo", 10,
"Pendiente", "juan123");

let reserva17 = new Reserva("Teatro Opera", "facundo", 10,
"Pendiente", "pedro123");

```



```

let reserva18= new Reserva("Tu Refuerso", "facundo", 10,
"Pendiente", "santiago123");

let reserva19= new Reserva("El Galpon", "facundo", 10, "Pendiente",
"agustin123");

let reserva20= new Reserva("Cafeteria Cafe", "facundo", 10,
"Pendiente", "sebastian123");

let reserva21= new Reserva("La Luz", "facundo", 10, "Pendiente",
"ezequiel123");


let reserva22 = new Reserva("Teatro Solis", "lucia", 10,
"Pendiente", "nicolas123");

let reserva23 = new Reserva("Museo del Carnaval", "lucia", 10,
"Pendiente", "juan123");

let reserva24 = new Reserva("Teatro Opera", "lucia", 10,
"Pendiente", "pedro123");

let reserva25= new Reserva("Tu Refuerso", "lucia", 10, "Pendiente",
"santiago123");

let reserva26= new Reserva("El Galpon", "lucia", 10, "Pendiente",
"agustin123");

let reserva27= new Reserva("Cafeteria Cafe", "lucia", 10,
"Pendiente", "sebastian123");

let reserva28= new Reserva("La Luz", "lucia", 10, "Pendiente",
"ezequiel123");


let reserva29 = new Reserva("Teatro Solis", "lucia", 10,
"Finalizada", "nicolas123", 4);

let reserva30 = new Reserva("Museo del Carnaval", "lucia", 10,
"Finalizada", "juan123", 4);

let reserva31 = new Reserva("Teatro Opera", "lucia", 10,
"Finalizada", "pedro123", 4);

```

```

let reserva32= new Reserva("Tu Refuerso", "lucia", 10,
"Finalizada", "santiago123", 4);

let reserva33= new Reserva("El Galpon", "lucia", 10, "Finalizada",
"agustin123", 4);

let reserva34= new Reserva("Cafeteria Cafe", "lucia", 10,
"Finalizada", "sebastian123", 4);

let reserva35= new Reserva("La Luz", "lucia", 10, "Finalizada",
"ezequiel123", 4);


let reserva36 = new Reserva("Teatro Solis", "maria", 10,
"Pendiente", "nicolas123");

let reserva37 = new Reserva("Museo del Carnaval", "maria", 10,
"Pendiente", "juan123");

let reserva38 = new Reserva("Teatro Opera", "maria", 10,
"Pendiente", "pedro123");

let reserva39= new Reserva("Tu Refuerso", "maria", 10, "Pendiente",
"santiago123");

let reserva40= new Reserva("El Galpon", "maria", 10, "Pendiente",
"agustin123");

let reserva41= new Reserva("Cafeteria Cafe", "maria", 10,
"Pendiente", "sebastian123");

let reserva42= new Reserva("La Luz", "maria", 10, "Pendiente",
"ezequiel123");


let reserva43 = new Reserva("Teatro Solis", "maria", 10,
"Finalizada", "nicolas123", 4);

let reserva44 = new Reserva("Museo del Carnaval", "maria", 10,
"Finalizada", "juan123", 4);

let reserva45 = new Reserva("Teatro Opera", "maria", 10,
"Finalizada", "pedro123", 4);

```

```

let reserva46= new Reserva("Tu Refuerso", "maria", 10,
"Finalizada", "santiago123", 4);

let reserva47= new Reserva("El Galpon", "maria", 10, "Finalizada",
"agustin123", 4);

let reserva48= new Reserva("Cafeteria Cafe", "maria", 10,
"Finalizada", "sebastian123", 4);

let reserva49= new Reserva("La Luz", "maria", 10, "Finalizada",
"ezequiel123", 4);

```

Locales Pre-Cargados:

Nombre / Usuario / Pass / TipoLocal / Cupomax / Foto / Estado / MaximoTotal

```

local1 = new Local("Teatro Solis", "Nicolas123", "Nicolas123",
"teatro", 50, "imgs/foto1.jpg", "habilitado", 100)

local2 = new Local("Museo del Carnaval", "Juan123", "Juan123",
"museo", 50, "imgs/foto2.jpg", "habilitado", 100)

local3 = new Local("Teatro Opera", "Pedro123", "Pedro123",
"teatro", 50, "imgs/foto3.jpg", "habilitado", 100)

local4 = new Local("Tu Refuerso", "Santiago123", "Santiago123",
"restaurant", 50, "imgs/foto4.jpg", "habilitado", 100)

local5 = new Local("El Galpon", "Agustin123", "Agustin123",
"restaurant", 50, "imgs/foto5.jpg", "habilitado", 100)

local6 = new Local("Cafeteria Cafe", "Sebastian123",
"Sebastian123", "restaurant", 50, "imgs/foto6.jpg", "habilitado",
100)

local7 = new Local("La Luz", "Ezequiel123", "Ezequiel123", "museo",
50, "imgs/foto7.jpg", "habilitado", 100)

```