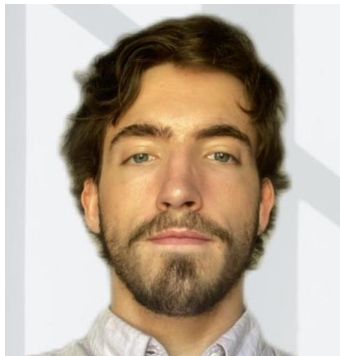


Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACION 1

DOCUMENTO DE ANÁLISIS



Martin Perez - 294829

M1C

Docente: Andrés de Sosa

Analista en Tecnologías de la Información

26/05/2022

Índice

1.	Descripción general del problema a resolver	3
1.1.	Tipos de usuario del sistema	3
1.2.	Listado de funcionalidades	3
2.	Detalle de Funcionalidades	4
2.1	F01 – Nombre de Funcionalidad 1	4
2.1.1	Acceso	4
2.1.2	Descripción	4
2.1.3	Interfaz de usuario	4
2.1.4	Validaciones	4

1. Descripción general del problema a resolver

Se espera crear una aplicación que permita a las personas acceder a funcionalidades relacionadas a la reserva de lugares en restaurantes, teatros o museos. De la misma forma la aplicación también deberá permitir el acceso a los usuarios locales los cuales tendrán otras funcionalidades distintas a la de los usuarios persona que podrán interaccionar con las reservas de los usuarios persona. Dichas funcionalidades se describirán mas adelante.

1.1. Tipos de usuario del sistema

- Usuario persona
- Usuario local

1.2. Listado de funcionalidades

Ejemplo:

F01 – Ingresar en la aplicación – Usuario/s: local y persona

F02 – Habilitar y deshabilitar la disponibilidad – Usuario/s: local

F03 – Cambiar el estado de las reservas – Usuario/s: local

F04 – Modificar el cupo máximo – Usuario/s: local

F05 – Visualizar información estadística (local) – Usuario/s: local

F06 – Registrarse en la aplicación – Usuario/s: persona

F07 – Realizar la solicitud de una reserva – Usuario/s: persona

F08 – Cancelar una reserva – Usuario/s: persona

F9 – Calificar una reserva finalizada – Usuario/s: persona

F10 – Ver listado de reservas pendientes– Usuario/s: persona

F11 – Ver información estadística (persona) – Usuario/s: 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 – Ingresar en la aplicación

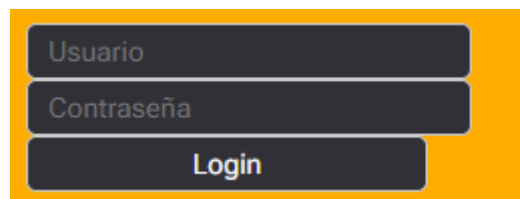
2.1.1 Acceso

Los locales y las personas pueden acceder a esta función

2.1.2 Descripción

Se trata de una pantalla de inicio donde los usuarios deberán digitar su usuario y contraseña de forma correcta pudiendo así acceder al menú del administrador. Los datos de ingreso ya estarán precargados.

2.1.3 Interfaz de usuario

La imagen muestra una interfaz de usuario para el login, consistiendo en tres campos de entrada y un botón. Los campos están etiquetados como 'Usuario', 'Contraseña' y 'Login'. El fondo es de color naranja.

2.1.4 Validaciones

El Administrador deberá digitar su usuario y su contraseña (ambos son obligatorios) y hacer clic en el botón ingresar. Si el usuario (el usuario será case insensitive) no está guardado en el sistema se mostrará:

- “El usuario ingresado no es válido”

Si el usuario existe, pero la contraseña (por el contrario, la contraseña será case sensitive) no coincide se muestra:

- “La contraseña no es correcta”

En caso de que si coincidan se deberá habilitar el menú de locales o de personas según el usuario que haya ingresado.

2.2F02 – Habilitar y deshabilitar la disponibilidad

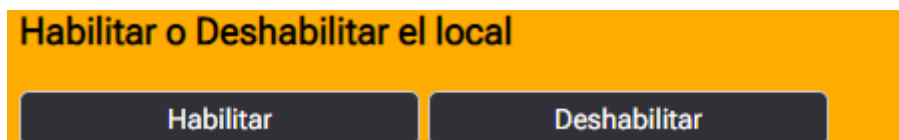
2.2.1 Acceso

Los locales pueden acceder a esta función

2.2.2 Descripción

El administrador podrá habilitar o deshabilitar a voluntad la disponibilidad de hacer reservas para un local. A su vez si el local llega a ocupar el máximo de cupos disponibles automáticamente se deshabilitará el local.

2.2.3 Interfaz de usuario



2.2.4 Validaciones

Para poder habilitar las reservas es necesario que existan cupos disponibles. En caso de que no queden cupos se deshabilitara la opción de hacer clic en dicho botón.

2.3F03 – Cambiar el estado de las reservas

2.3.1 Acceso

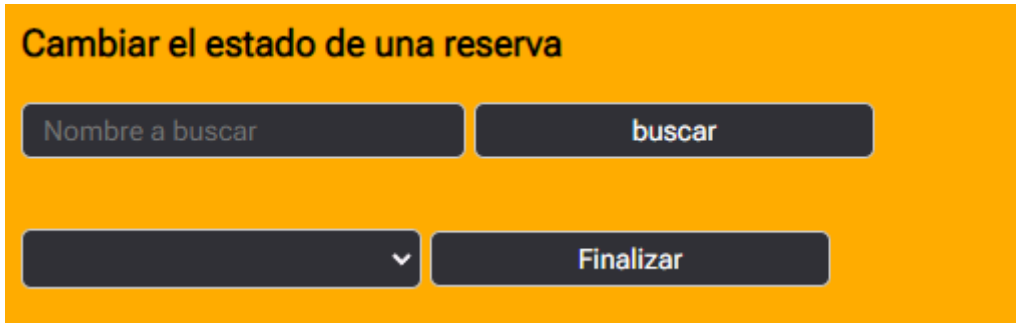
Los locales pueden acceder a esta función

2.3.2 Descripción

Se permitirá cambiar el estado de las reservas de “pendiente” a “finalizada” (de forma individual). A su vez también se contará con un buscador que permita encontrar reservas a partir del nombre del cliente entre el total de las reservas pendientes. El buscador

permitirá encontrar nombres mencionando tan solo una fracción de este, es decir no es necesario que se digite el nombre completo para encontrarlo.

2.3.3 Interfaz de usuario



The screenshot shows a user interface with an orange background. At the top, the title "Cambiar el estado de una reserva" is displayed in black. Below the title, there are two input fields: the first contains the placeholder text "Nombre a buscar" and the second contains the text "buscar". Below these fields, there is a dropdown menu with a downward arrow and a button labeled "Finalizar".

2.3.4 Validaciones

Es obligatorio que se digite al menos una fracción del nombre para buscar el nombre. Si lo digitado no es encontrado en los datos se deberá mostrar un mensaje que indique:

- “No hay clientes con ese nombre que tengan una reserva pendiente”

2.4F04 – Modificar cupo máximo

2.4.1 Acceso

Los locales pueden acceder a esta función

2.4.2 Descripción

El local podrá modificar el cupo máximo del mismo.

2.4.3 Interfaz de usuario



The screenshot shows a user interface with an orange background. It contains two input fields: the first contains the placeholder text "Digitar nuevo cupo maximo" and the second contains the text "Cambiar".

2.4.4 Validaciones

En el campo de texto es obligatorio insertar un numero natural (mayor que 0) y además se debe verificar que no haya reservas en estado pendiente.

En los posibles casos de error se indicarán los siguientes mensajes según corresponda:

- “Digitar un número mayor que 0”
- “Hay una reserva pendiente, no se puede modificar el cupo máximo”

2.5F05 – Visualizar información estadística (local)

2.5.1 Acceso

Los locales pueden acceder a esta función

2.5.2 Descripción

Esta función permite que el administrador pueda ver el porcentaje de ocupación del lugar en relación con el cupo máximo y a las reservas pendientes. A su vez también se mostrará el promedio de las calificaciones de su local y también de los otros locales registrados en la base de datos. También se mostrará la cantidad de reservas pendientes y finalizadas que tiene el local.

2.5.3 Interfaz de usuario

Informacion Estadistica

Porcentaje de ocupacion: 0%

Promedio de calificaciones: Aun no hay calificaciones

Local	Promedio de Calificaciones
Francis	No hay calificaciones
Hard Rock	No hay calificaciones
Gourmet	No hay calificaciones
Louvre	No hay calificaciones
Mucha Arte	No hay calificaciones
Shakespeare	No hay calificaciones
Grecia	No hay calificaciones

Reservas pendientes: 0

Reservas finalizadas: 3

2.5.4 Validaciones

No se realizarán validaciones ya que no se debe contemplar ningún ingreso de datos del local.

2.6F06 – Registrarse en la aplicación

2.6.1 Acceso

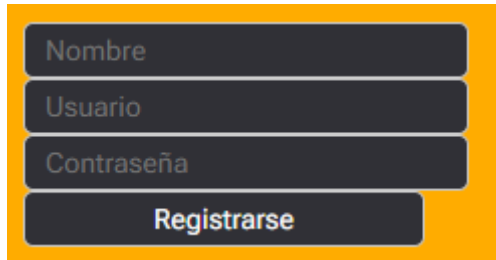
Los usuarios persona pueden acceder a esta función

2.6.2 Descripción

El usuario deberá ingresar un nombre, usuario y una contraseña. El usuario no puede coincidir con otro usuario ya registrado (incluyendo los usuarios de los locales) y a su vez

la contraseña deberá tener un mínimo de 6 caracteres, una mayúscula, una minúscula y un número.

2.6.3 Interfaz de usuario

Una interfaz de usuario para el registro de usuarios. Se trata de un formulario con un fondo naranja. Contiene cuatro campos de entrada de texto con un fondo gris oscuro y texto gris claro: 'Nombre', 'Usuario', 'Contraseña' y un botón 'Registrarse'.

2.6.4 Validaciones

Todos los datos son obligatorios. El usuario deberá ingresar el nombre, usuario y la contraseña.

Si no se completan se mostrará:

- “Completar todos los campos”

El usuario debe ser único, no puede estar repetido, en caso de error se indicará:

- “El usuario elegido ya existe, por favor elija otro”

La contraseña deberá tener mínimo 6 caracteres, una mayúscula, una minúscula y un número, en caso de error se indicará:

- “La contraseña seleccionada no cumple con los requisitos mínimos (6 caracteres, al menos una mayúscula, una minúscula y un numero)”

2.7F07 – Realizar la solicitud de una reserva

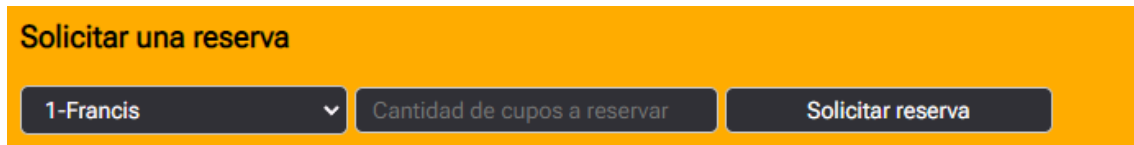
2.7.1 Acceso

Los usuarios normales pueden acceder a esta función

2.7.2 Descripción

El usuario tendrá la opción de indicar la cantidad de personas y el local y realizar una reserva. Solo se procesará la reserva si la persona no tiene otras reservas en estado pendiente en ese local. La reserva se creará en estado “pendiente”.

2.7.3 Interfaz de usuario



Solicitar una reserva

1-Francis ▼ Cantidad de cupos a reservar Solicitar reserva

2.7.4 Validaciones

Se debe validar que la persona no tenga reservas previas pendientes en ese local, que digite un numero natural y que haya suficientes cupos disponibles para procesar esa reserva. En caso de error mostrar los siguientes mensajes según corresponda:

- “Digitar un número mayor a 0”
- “Ya se ha solicitado una reserva”

Si no quedan cupos disponibles entonces no aparecerá el local entre la lista de opciones ya que estará deshabilitado.

2.8F08 – Cancelar una reserva

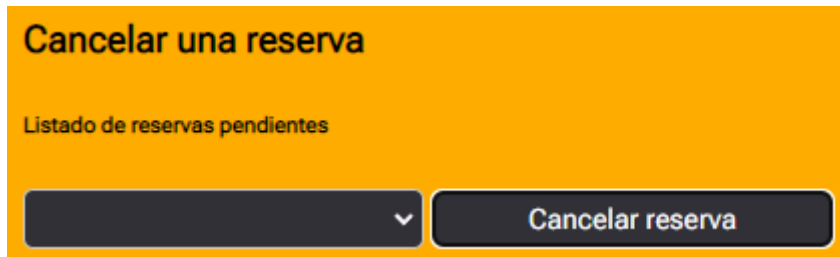
2.8.1 Acceso

Los usuarios persona pueden acceder a esta función

2.8.2 Descripción

El usuario podrá ver sus solicitudes de reserva en estado “pendiente” y podrá pasar las mismas a estado “cancelada”.

2.8.3 Interfaz de usuario



2.8.4 Validaciones

Para poder cancelar la reserva la misma se debe encontrar en estado pendiente. Una vez cancelada se eliminara de la lista dentro del select.

2.9F9 – Calificar una reserva finalizada

2.9.1 Acceso

Los usuarios persona pueden acceder a esta función

2.9.2 Descripción

El usuario podrá ver sus reservas finalizadas, seleccionar de a una y calificarlas del 1 al 5

2.9.3 Interfaz de usuario



2.9.4 Validaciones

El usuario debe digitar un numero entre 1 y 5. Si digita otro valor se mostrará:

- “Digitar una calificación valida”

2.10 F10 – Ver listado de reservas pendientes

2.10.1 Acceso

Los usuarios persona pueden acceder a esta función

2.10.2 Descripción

El usuario podrá ver un listado en el que se muestren las reservas pendientes. Las mismas mostrarán la foto del local, el nombre y la cantidad de cupos solicitados.

2.10.3 Interfaz de usuario



2.10.4 Validaciones

Para poder mostrar las reservas pendientes las mismas deben existir. En caso de que no haya reservas pendientes se indicara:

- “No hay reservas pendientes”

2.11 F11 –Ver información estadística (persona)

2.11.1 Acceso

Los usuarios persona pueden acceder a esta función

2.11.2 Descripción

Se le mostrara al usuario persona un listado que aparezca:

- Locales donde el cliente haya hecho reservas que haya completado
- Cantidad de reservas que hizo en dicho local
- El total de reservas que tiene el local (se contemplan todas las reservas en todos los estados)
- El porcentaje de reservas que corresponden a ese usuario en ese local.

También se mostrará el local en el que el usuario haya hecho mas reservas, y en caso de ser mas de uno se muestran todos (para este cálculo se contemplan todas las reservas en todos los estados)

2.11.3 Interfaz de usuario

Informacion Estadistica

Reservas finalizadas	Reservas completadas	Reservas del local	Q reservada/Reservas del local
Francis	3	5	60%

Local/es mas reservado/s: Francis, con 3 reservas

2.11.4 Validaciones

No se realizarán validaciones ya que no se debe contemplar ningún ingreso de datos del usuario persona.

INFORME DE TESTING**CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIADAS A LOS USUARIOS LOCAL Y PERSONA:****F-01: Ingresar en la aplicación**

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F01-T01	Login correcto persona	Digita usuario y contraseña guardados en la lista de usuarios persona		Se muestra el menú para las personas	Se muestra el menú para las personas	P
F01-T02	Login correcto local	Digita usuario y contraseña guardados en la lista de usuarios locales		Se muestra el menú para los locales	Se muestra el menú para los locales	P
F01-T03	Usuario incorrecto	Digita un usuario que no esta guardado en ninguna de las dos listas		Se muestra "usuario no existe"	Se muestra "usuario no existe"	P
F01-T04	Contraseña incorrecta	Digita un usuario que esta guardado en la lista Digita una contraseña que no coincide con el usuario		Se muestra "Contraseña incorrecta"	Se muestra "Contraseña incorrecta"	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIADAS AL USUARIO LOCAL:**F-02: Habilitar y deshabilitar disponibilidad**

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F02-T01	Se deshabilita el local	Hacer clic en disponibilidad. Hacer clic en deshabilitar		Se muestra "local deshabilitado" y se deshabilita el local para los usuarios persona	Se muestra "local deshabilitado" y se deshabilita el local para los usuarios persona	P
F02-T02	Se habilita el local	Hacer clic en disponibilidad. Hacer clic en habilitar		Se "muestra local habilitado" y se habilita el local para los usuarios persona	Se "muestra local habilitado" y se habilita el local para los usuarios persona	P
F02-T03	Se deshabilita automáticamente el local	Un usuario persona reserva todos los cupos disponibles.		Se muestra "local deshabilitado" y se deshabilita el local para los usuarios persona. Se deshabilita el botón "habilitar"	Se muestra "local deshabilitado" y se deshabilita el local para los usuarios persona. Se deshabilita el botón "habilitar"	P

CASOS DE PRUEBA - PROGRAMACIÓN I – OBLIGATORIO MP 294829

F-03: Cambiar estado de las reservas

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F03-T01	Se busca una reserva y se finaliza	Se digita un nombre existente en la casilla "nombre a buscar" y se hace clic en buscar. Se selecciona en el select y se hace clic en finalizar		El usuario buscado con la reserva pendiente aparece en el select. Cuando se hace clic en finalizar desaparece del select y se pasa la reserva a estado finalizado	El usuario buscado con la reserva pendiente aparece en el select. Cuando se hace clic en finalizar desaparece del select y se pasa la reserva a estado finalizado	P
F03-T02	Se busca una reserva, pero no coincide con ningún resultado	Se digita un nombre no existente en la casilla "nombre a buscar" y se hace clic en buscar.		Se muestra "no hay clientes con ese nombre que tengan una reserva pendiente" Se muestra el select vacío	Se muestra "no hay clientes con ese nombre que tengan una reserva pendiente" Se muestra el select vacío	P

F-04: Modificar cupo máximo

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F04-T01	Cambio correcto	Se digita un numero valido y se hace clic en cambiar		Se muestra "Se ha modificado el cupo máximo" y "El cupo máximo actual es de x lugares" y se cambia el cupo máximo	Se muestra "Se ha modificado el cupo máximo" y "El cupo máximo actual es de x lugares" y se cambia el cupo máximo	P
F04-T02	Numero invalido	Se digita un numero invalido y se hace clic en cambiar		Se muestra "Digitar un número mayor que 0" No se cambia el cupo máximo	Se muestra "Digitar un número mayor que 0" No se cambia el cupo máximo	P
F04-T03	Existe una reserva pendiente	Existe previamente una reserva pendiente para ese local. Se digita un numero Se hace clic en cambiar		Se muestra "Hay una reserva pendiente, no se puede modificar el cupo máximo" No se modifica el cupo máximo	Se muestra "Hay una reserva pendiente, no se puede modificar el cupo máximo" No se modifica el cupo máximo	P

F-05: Visualizar información estadística (Local)

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F05-T01	Se muestra correctamente la información estadística	Se ingresa como usuario local y se hace clic en Información estadística en la lista de menús.		Se muestran correctamente los datos del local logueado	Se muestran correctamente los datos del local logueado	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIADAS AL USUARIO PERSONA:

F-06: Registrarse en la aplicación

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F06-T01	Se registra correctamente	Se ingresa un nombre Se ingresa un usuario que no esté registrado Se ingresa una contraseña que cumpla con los requisitos Se hace clic en registrarse.		Se muestra "registro exitoso" Se registra al usuario persona	Se muestra "registro exitoso" Se registra al usuario persona	P
F06-T02	No completa todos los campos	No se llena alguno de los campos y se hace clic en registrarse.		Se muestra "completar todos los campos" No se registra el usuario	Se muestra "completar todos los campos" No se registra el usuario	P
F06-T03	Digita un usuario ya registrado	Se ingresa un nombre Se ingresa una contraseña Se ingresa un usuario ya existente Se hace clic en registrarse		Se muestra "El usuario elegido ya existe, por favor elija otro" No se registra el usuario	Se muestra "El usuario elegido ya existe, por favor elija otro" No se registra el usuario	P
F06-T04	Digita una contraseña que no cumple con todos los requisitos	Se ingresa un nombre Se ingresa un usuario que no este registrado Se ingresa una contraseña que no cumple con los requisitos Se hace clic en registrar		Se muestra "la contraseña seleccionada no cumple con los requisitos mínimos" No se registra el usuario	Se muestra "la contraseña seleccionada no cumple con los requisitos mínimos" No se registra el usuario	P

F-07: Realizar la solicitud de una reserva

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F07-T01	Se realiza una reserva	Se selecciona un local en el select. Se digita un numero de reservas mayor a 0. Se hace clic en "solicitar reservas".		Se muestra "Se han solicitado x cupos". Se crea la reserva	Se muestra "Se han solicitado x cupos". Se crea la reserva	P
F07-T02	Se digita un número menor a 1	Se selecciona un local en el select. Se digita un número menor a 0. Se hace clic en "solicitar reservas"		Se muestra "Digitar un número mayor a 0". No se crea la reserva	Se muestra "Digitar un número mayor a 0". No se crea la reserva	P
F07-T03	Se hace una reserva en un local donde ya hay una reserva pendiente	Se selecciona un local donde el usuario ya tiene una reserva pendiente. Se digita un número. Se hace clic en "solicitar reservas"		Se muestra "Ya tiene una reserva pendiente en ese local" No se crea la reserva.	Se muestra "Ya tiene una reserva pendiente en ese local" No se crea la reserva.	P

F-08: Cancelar una reserva

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F08-T01	Se cancela una reserva exitosamente	Se hace clic en el select y se elige el local del cual se quiere cancelar la reserva pendiente. Se hace clic en cancelar		Se borra el local del select. Se cancela la reserva. Se muestra "se ha cancelado la reserva.	Se borra el local del select. Se cancela la reserva. Se muestra "se ha cancelado la reserva.	P

F-09: Calificar una reserva finalizada

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F09-T01	Se califica exitosamente una reserva finalizada	Se elige en el select el local a calificar. Se digita un numero entre 1 y 5 como calificación. Se hace clic en "calificar"		Se muestra "Calificación agregada correctamente". Se agrega la calificación. Se borra la reserva finalizada del select.	Se muestra "Calificación agregada correctamente". Se agrega la calificación. Se borra la reserva finalizada del select.	P
F09-T02	Se digita un numero invalido como calificación	Se elige un local en el select. Se digita un numero que no esta entre 1 y 5. Se hace clic en "calificar"		Se muestra "Digitar una calificación valida". No se agrega la calificación. Se sigue mostrando el local en el select.	Se muestra "Digitar una calificación valida". No se agrega la calificación. Se sigue mostrando el local en el select.	P

F-10: Ver listado de reservas pendientes

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F10-T01	Hay reservas pendientes	Se hace clic en reservas pendientes habiendo reservas pendientes.		Se muestra la lista de reservas pendientes incluyendo: Nombre del local, foto del local y cantidad de cupos reservados	Se muestra la lista de reservas pendientes incluyendo: Nombre del local, foto del local y cantidad de cupos reservados	P
F10-T02	No hay reservas pendientes	Se hace clic en reservas pendientes no habiendo reservas pendientes.		Se muestra "no hay reservas pendientes"	Se muestra "no hay reservas pendientes"	P

CASOS DE PRUEBA - PROGRAMACIÓN I – OBLIGATORIO MP 294829

F-11: Ver información estadística

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F11-T01	Se muestra correctamente la información estadística	Se ingresa como usuario local y se hace clic en Información estadística en la lista de menús.		Se muestra una tabla con los nombres de los locales de las reservas finalizadas, la cantidad de reservas finalizadas para ese local, las reservas del local en total incluyendo todas las existentes independientemente del estado, el porcentaje que representa la cantidad de reservas completadas de ese usuario para ese local en relación con el total de reservas de ese local. Se muestra el o los locales donde el usuario hizo mas reservas y el numero de las mismas.	Se muestra una tabla con los nombres de los locales de las reservas finalizadas, la cantidad de reservas finalizadas para ese local, las reservas del local en total incluyendo todas las existentes independientemente del estado, el porcentaje que representa la cantidad de reservas completadas de ese usuario para ese local en relación con el total de reservas de ese local. Se muestra el o los locales donde el usuario hizo más reservas y el número de estas.	P

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>Document</title>
  <script src="js/codigo.js"></script>
  <script src="js/class.js"></script>
  <link rel = "stylesheet" href="css/style.css">
</head>
<body>

  <!-- Login -->
  <div id="mnuLogin">
    <input type="text" id="txtUserLogin" placeholder="Usuario"><br>
    <input type="password" id="txtPassLogin" placeholder="Contraseña"><br>
    <input type="button" id="btnLogin" value="Login">
    <p id="msjLogin"></p>
  </div>

  <!-- Registrarse -->
  <div id="mnuRegistrar">
    <input type="text" id="txtNameRegis" placeholder="Nombre"><br>
    <input type="text" id="txtUserRegis" placeholder="Usuario"><br>
    <input type="text" id="txtPassRegis" placeholder="Contraseña"><br>
    <input type="button" id="btnRegis" value="Registrarse">
    <p id="msjRegis"></p>
  </div>

  <!------- LOCAL ----->
  <!-- Menu Local -->
  <div id="cabezal" class="clearfix">
    <ul id="mnuLocal">
      <li>
        <a id="disponibilidad">Disponibilidad</a>
      </li>
      <li>
        <a id="estadoReservas">Ver estado de reservas</a>
      </li>
      <li>
```

```

        <a id="cupoMaximo">Cupo maximo</a>
    </li>
    <li>
        <a id="infoEstadisticaLocales">Informacion estadistica</a>
    </li>
    <li>
        <a id="cerrarSesionL">Cerrar sesion</a>
    </li>
</ul>
</div>

<div id="mnuDisponibilidad">
    <h3>Habilitar o Deshabilitar el local</h3>
    <h6>ATENCION: Si se llenan los cupos hasta el cupo maximo no se podra
volver a habilitar el local</h6>
    <input type="button" value="Habilitar" id="btnHabilitar">
    <input type="button" value="Deshabilitar" id="btnDeshabilitar">
    <p id="msjDisponibilidad">Local Habilitado</p>
</div>

<div id="mnuEstadoReservas">
    <h3>Cambiar el estado de una reserva</h3>
    <input type="text" id="txtBuscarNombre" placeholder="Nombre a buscar">
    <input type="button" id="btnBuscar" value="buscar"><br><br>
    <p id="msjBusqueda"></p>
    <select id="sctReservaACambiar"></select>
    <input type="button" value="Finalizar" id="btnFinalizar">
    <p id="msjEstadoReservas"></p>
</div>

<div id="mnuCupoMaximo">
    <h3>Cambiar cupo maximo</h3>
    <p id="msjCupoMaximoActual"></p>
    <input type="number" id="txtCupoMaximoNuevo" placeholder="Digitar nuevo
cupo maximo">
    <input type="button" value="Cambiar" id="btnCambiarCupoMaximo">
    <p id="msjCupoMaximo"></p>
</div>

<div id="mnuInfoEstadisticaLocales">
    <h3>Informacion Estadistica</h3>
    <p id="porcentajeOcupacion"></p>
    <p id="promedioCalificacion"></p>

    <div id="tablaCalificaciones">

```

```

    </div>

    <p id="reservasPendientesLocal"></p>
    <p id="reservasFinalizadasLocal"></p>

</div>

<!------- PERSONA ----->

    <!-- Menu Persona -->

<div id="cabezal" class="clearfix">
    <div>
        <ul id="mnuPersona">

            <li>
                <a id="solicitarReserva">Solicitar reserva</a>
            </li>
            <li>
                <a id="calificarReserva">Calificar reserva</a>
            </li>
            <li>
                <a id="reservasPendientes">Reservas pendientes</a>
            </li>
            <li>
                <a id="infoEstadisticaPersonas">Informacion estadistica</a>
            </li>
            <li>
                <a id="cerrarSesionP">Cerrar sesion</a>
            </li>
        </ul>
    </div>
</div>

<div id="mnuSolicitarReserva">
    <h3>Solicitar una reserva</h3>
    <select id="sctLocalAReservar"></select>
    <input type="number" id="txtCantidadAReservar" placeholder="Cantidad de cupos
a reservar">
    <input type="button" id="btnReservar" value="Solicitar reserva">
    <p id="msjReserva"></p>

<hr>

```

```

    <h3>Cancelar una reserva</h3>
    <h6>Listado de reservas pendientes</h6>
    <select id="sctLocalACancelar"></select>
    <input type="button" id="btnCancelarReserva" value="Cancelar reserva">
    <p id="msjCancelarReserva"></p>
</div>

<div id="mnuCalificarReserva">
    <h3>Calificar una reserva finalizada</h3>
    <h5>Digitar una calificacion entre 1 y 5</h5>
    <select id="sctLocalACalificar"></select>
    <input type="number" id="txtCalificacion" placeholder="Calificacion">
    <input type="button" id="btnCalificar" value="Calificar">
    <p id="msjAgregarCalificacion"></p>
</div>

<div id="mnuReservasPendientes">
    <h1>Reservas Pendientes</h1>
    <div id="lstReservasPendientes">

    </div>
    <p id="msjReservasPendientes"></p>
</div>

<div id="mnuInfoEstadisticaPersonas">
    <h3>Informacion Estadistica</h3>
    <div id="tablaReservas">
    </div>
    <p id="mayorQReservas"></p>
</div>

</body>
</html>

```

JavaScript – código.js

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

function inicio(){
    document.querySelector("#btnLogin").addEventListener("click", login);
    document.querySelector("#btnRegis").addEventListener("click", registrarse);
    document.querySelector("#cerrarSesionL").addEventListener("click", cerrarSesion);
    document.querySelector("#cerrarSesionP").addEventListener("click", cerrarSesion);
    document.querySelector("#disponibilidad").addEventListener("click", disponibilidad);
    document.querySelector("#btnHabilitar").addEventListener("click", habilitar);
    document.querySelector("#btnDeshabilitar").addEventListener("click", deshabilitar);
    document.querySelector("#solicitarReserva").addEventListener("click", mnuSolicitarReserva);
    document.querySelector("#btnReservar").addEventListener("click", reservar);
    document.querySelector("#btnCancelarReserva").addEventListener("click", cancelarReserva);
    document.querySelector("#calificarReserva").addEventListener("click", mnuCalificarReserva);
    document.querySelector("#btnCalificar").addEventListener("click", calificarReservaFinalizada);
    document.querySelector("#reservasPendientes").addEventListener("click", mnuReservasPendientes);
    document.querySelector("#estadoReservas").addEventListener("click", mnuEstadoReservas);
    document.querySelector("#btnFinalizar").addEventListener("click", finalizarReserva);
    document.querySelector("#cupoMaximo").addEventListener("click", mnuCupoMaximo);
    ;
    document.querySelector("#btnCambiarCupoMaximo").addEventListener("click", cambiarCupoMaximo);
    document.querySelector("#infoEstadisticaLocales").addEventListener("click", mnuInfoEstadisticaLocales);
    document.querySelector("#infoEstadisticaPersonas").addEventListener("click", mnuInfoEstadisticaPersonas);
    document.querySelector("#btnBuscar").addEventListener("click", buscarNombre);

    ocultarTodo();
    precargarUsuariosLocales();
    precargarUsuariosPersona();
}
```



```

    precargarReservas();
}

//Esta funcion borra los mensajes de cada menu
function borrarMensajes(){
    document.querySelector("#msjLogin").innerHTML=""
    document.querySelector("#msjRegis").innerHTML=""
    document.querySelector("#msjBusqueda").innerHTML=""
    document.querySelector("#msjEstadoReservas").innerHTML=""
    document.querySelector("#msjCupoMaximo").innerHTML=""
    document.querySelector("#msjCupoMaximoActual").innerHTML=""
    document.querySelector("#msjReserva").innerHTML=""
    document.querySelector("#msjCancelarReserva").innerHTML=""
    document.querySelector("#msjAgregarCalificacion").innerHTML=""
    document.querySelector("#msjReservasPendientes").innerHTML=""
}

function ocultarTodo(){
    document.querySelector("#mnuLocal").style.display="none";
    document.querySelector("#mnuPersona").style.display="none";
    ocultarFuncionLocal();
    ocultarFuncionPersona();
    borrarMensajes();
}

//Esta funcion oculta cualquier opcion que haya abierto el usuario local
function ocultarFuncionLocal(){
    borrarMensajes();
    document.querySelector("#mnuDisponibilidad").style.display="none";
    document.querySelector("#mnuEstadoReservas").style.display="none";
    document.querySelector("#mnuCupoMaximo").style.display="none";
    document.querySelector("#mnuInfoEstadisticaLocales").style.display="none";
}

//Esta funcion oculta cualquier opcion que haya abierto el usuario persona
function ocultarFuncionPersona(){
    borrarMensajes();
    document.querySelector("#mnuSolicitarReserva").style.display="none";
    document.querySelector("#mnuCalificarReserva").style.display="none";
    document.querySelector("#mnuReservasPendientes").style.display="none";
    document.querySelector("#mnuInfoEstadisticaPersonas").style.display="none";
}

```

```

}

// ----- LOGIN -----
let listaLocales=[];
let posUsuarioLogueado=-1;//esta variable me dice la posicion del usuario que se
logueo dentro de la lista personas o locales

function login(){
    let userSensitive = document.querySelector("#txtUserLogin").value;
    let user = userSensitive.toLowerCase();//Lo que digita el usuario pasado a
minusculas para que sea case insensitive
    let pass = document.querySelector("#txtPassLogin").value;
    let posUser = existeUsuario(user); //Me da la posicion del usuario (si
existe, sino =-1)
    let tipo = tipoUsuario(user); //Me dice si el usuario es local(0) o
persona(1)
    let coincide=false;

    if (posUser!=-1){//Esa condicion significa "si el usuario existe"
        if(tipo==0){
            coincide=contraseñaCoincideL(pass,posUser) //Esta funcion invocada me
retorna true si la contraseña coincide con el usuario si es local
        }
        if(tipo==1){
            coincide=contraseñaCoincideP(pass,posUser) //Esta funcion invocada me
retorna true si la contraseña coincide con el usuario si es persona
        }
    }
    else{
        document.querySelector("#msjLogin").innerHTML="Usuario no existe"
    }

    if(coincide){
        if(tipo==0){
            document.querySelector("#mnuLocal").style.display="block"; //si coincide
y es local se despliega el menu para Locales
            document.querySelector("#mnuLogin").style.display="none";
            document.querySelector("#mnuRegistrar").style.display="none";

            //El for a continuacion guarda en la variable "posUsuarioLogueado" la
posicion del usuario (Local) que se logueo, esta posicion
            //me va a ser util mas adelante en otras funciones.
            for(let pos0=0;pos0<listaLocales.length;pos0++){
                let posicionUsuario=listaLocales[pos0]
            }
        }
    }
}

```

```

        if(posicionUsuario.usuario==user){
            posUsuarioLogueado=pos0
        }
    }
}

if(tipo==1){
    document.querySelector("#mnuPersona").style.display="block"; //si
coincide y es persona se despliega el menu para personas
    document.querySelector("#mnuLogin").style.display="none";
    document.querySelector("#mnuRegistrar").style.display="none";

    //El for a continuacion guarda en la variable "posUsuarioLogueado" la
posicion del usuario (Persona) que se logueo, esta posicion
//me va a ser util mas adelante en otras funciones.
    for(let pos1=0;pos1<listaPersonas.length;pos1++){
        let posicionUsuario=listaPersonas[pos1]
        if(posicionUsuario.usuario==user){
            posUsuarioLogueado=pos1
        }
    }
}

}

document.querySelector("#txtUserLogin").value=""
document.querySelector("#txtPassLogin").value=""
}

function existeUsuario(user){

    //Esta funcion me dice si ya existe ese nombre de usuario ingresado tanto en
Locales como en Personas
    // si existe retorna la posicion en que se encuentra en el array, sino
retorna -1

    let posUser=-1
    for(pos=0;pos<listaLocales.length;pos++){
        if(listaLocales[pos].usuario.toLowerCase()==user){
            posUser=pos
        }
    }
    for(pos=0;pos<listaPersonas.length;pos++){
        if(listaPersonas[pos].usuario.toLowerCase()==user){
            posUser=pos
        }
    }
}

```

```

    }
  }
  return posUser
}

function tipoUsuario(user){

  //Esta funcion me dice si este usuario fue encontrado en la lista de locales
  (0) o de personas (1)

  let tipo=0
  for(pos=0;pos<listaLocales.length;pos++){
    if(listaLocales[pos].usuario==user){
      tipo=0
    }
  }
  for(pos=0;pos<listaPersonas.length;pos++){
    if(listaPersonas[pos].usuario==user){
      tipo=1
    }
  }
  return tipo
}

function contraseñaCoincideL(pass,posUser){
  let coincide=false
  if(listaLocales[posUser].contraseña==pass){//este if me dice si el
usuario y la contra coinciden si el usuario es local
    coincide=true
  }
  else{
    document.querySelector("#msjLogin").innerHTML="Contraseña incorrecta">//Si
no coincide se muestra este mensaje
  }
  return coincide
}

function contraseñaCoincideP(pass,posUser){
  let coincide=false
  if(listaPersonas[posUser].contraseña==pass){//este if me dice si el
usuario y la contra coinciden si el usuario es persona
    coincide=true
  }
  else{

```

```

        document.querySelector("#msjLogin").innerHTML="Contraseña
incorrecta">//Si no coincide se muestra este mensaje
    }
    return coincide
}

// ----- REGISTRAR PERSONA -----
-
let listaPersonas=[];

function registrarse(){
let nameP = document.querySelector("#txtNameRegis").value;
let userP = document.querySelector("#txtUserRegis").value;
let passP = document.querySelector("#txtPassRegis").value;

if(nameP==" "||userP==" "||passP==""){
    document.querySelector("#msjRegis").innerHTML="Completar todos los campos"
//Si el usuario no llena todos los campos no se puede registrar
}
else{
    if(existeUsuario(userP)==-1){ //el usuario no debe existir
        if(contraseñaValida(passP)){
            registrarPersona(nameP,userP,passP);
            document.querySelector("#msjRegis").innerHTML="Registro exitoso"
            document.querySelector("#txtNameRegis").value=""
            document.querySelector("#txtPassRegis").value=""
            document.querySelector("#txtUserRegis").value=""

        }
        else{
            document.querySelector("#msjRegis").innerHTML="La contraseña
seleccionada no cumple con los requisitos minimos (6 caracteres, al menos una
mayúscula, una minúscula y un numero)"
            document.querySelector("#txtPassRegis").value=""

        }
    }
    else{
        document.querySelector("#msjRegis").innerHTML="El usuario elegido ya
existe, por favor elija otro"
        document.querySelector("#txtUserRegis").value=""
        document.querySelector("#txtPassRegis").value=""
    }
}
}
}

```

```
function contraseñaValida(passP){
    let valida=false;
    let mayus=false;
    let minus=false;
    let numero=false;
    let cantidadCaracteres=false;

    //La contraseña debe tener mas de 6 caracteres
    if(passP.length>5){
        cantidadCaracteres=true
    }

    //La contraseña debe tener al menos una mayuscula
    for(let pos=0;pos<passP.length;pos++){
        let letra=passP.charAt(pos);
        if(letra==letra.toUpperCase()){
            mayus=true
        }
    }

    //La contraseña debe tener al menos una minuscula
    for(let pos1=0;pos1<passP.length;pos1++){
        let letra1=passP.charAt(pos1);
        if(letra1==letra1.toLowerCase()){
            minus=true
        }
    }

    //La contraseña debe tener al menos un numero
    for(let pos3=0;pos3<passP.length;pos3++){
        let letra3=passP.charAt(pos3);
        if(letra3=="0"||letra3=="1"||letra3=="2"||letra3=="3"||letra3=="4"||letra3=="5"||letra3=="6"||letra3=="7"||letra3=="8"||letra3=="9"){
            numero=true
        }
    }

    //si se cumplen todas las anteriores la variable "valida" retorna true
    if(mayus&&minus&&numero&&cantidadCaracteres){
        valida=true;
    }
    return valida;
}
```

```

function registrarPersona(nombre,usuario,contraseña){
    let objPersona=new persona(nombre,usuario,contraseña);
    listaPersonas.push(objPersona)
}

// ----- P R E C A R G A   D E   D A T O S -----
-----

// --- Precarga de Locales --

function precargarUsuariosLocales(){
    agregarLocales("Francis","rest1","123","restaurant","Av wilson
12",5,"fotos/foto4.jpg",true,0);
    agregarLocales("Hard Rock","rest2","223","restaurant","Av Italia
720",3,"fotos/foto5.jpg",true,0);
    agregarLocales("Gourmet","rest3","323","restaurant","Maximo Tajes
56",10,"fotos/foto6.jpg",true,0);
    agregarLocales("Louvre","mus1","113","museo","Av Uruguay
2666",2,"fotos/foto1.jpg",true,0);
    agregarLocales("Mucha Arte","mus2","123","museo","Valdivia
16",8,"fotos/foto2.jpg",true,0);
    agregarLocales("Shakespeare","tea1","121","teatro","Divina Comedia
134",3,"fotos/foto8.jpg",true,0);
    agregarLocales("Grecia","tea2","122","teatro","Portezuelo
8",4,"fotos/foto3.jpg",true,0);
}

function
agregarLocales(nombre,usuario,contraseña,tipoLocal,direccion,cupoMaximo,foto,esta
do,cuposOcupados){
    let objLocal=new
local(nombre,usuario,contraseña,tipoLocal,direccion,cupoMaximo,foto,estado,cuposO
cupados)
    listaLocales.push(objLocal)
}

// --- Precarga de Personas --

function precargarUsuariosPersona(){
    agregarPersonas("Martin","user1","123");
    agregarPersonas("Martina","user2","123");
    agregarPersonas("Pedro","user3","123");
    agregarPersonas("Juan","user4","123");
}

```

```

    agregarPersonas("Santiago","user5","123");
}

function agregarPersonas(nombre,usuario,contraseña){
    let objPersona=new persona(nombre,usuario,contraseña)
    listaPersonas.push(objPersona)
}

// --- Precarga de reservas ---

function precargarReservas(){
    agregarReservas(0,"Francis",3,"finalizada")
    agregarReservas(0,"Grecia",2,"pendiente")
    agregarReservas(1,"Hard Rock",4,"finalizada")
    agregarReservas(1,"Louvre",1,"pendiente")
    agregarReservas(2,"Shakespeare",2,"pendiente")
}

function agregarReservas(pos,localElegido,cantidadReservada,estadoReserva){
    let objReserva=new reserva(localElegido,cantidadReservada,estadoReserva)
    listaPersonas[pos].listaReservas.push(objReserva)
}

// ----- Cerrar Sesion -----

//Esta funcion cerrar sesion oculta los menus de persona o de local y muestra
nuevamente el login

function cerrarSesion(){
    document.querySelector("#mnuLocal").style.display="none";
    document.querySelector("#mnuPersona").style.display="none";
    document.querySelector("#mnuLogin").style.display="block";
    document.querySelector("#mnuRegistrar").style.display="block";
    ocultarFuncionLocal();
    ocultarFuncionPersona();
}

// ----- FUNCIONES LOCAL -----

```



```

// ----- Habilitar y deshabilitar disponibilidad -----
-----

function disponibilidad(){
    ocultarFuncionLocal();
    document.querySelector("#mnuDisponibilidad").style.display="block";
    deshabilitarBtnHabilitar();
}

function habilitar(){
    listaLocales[posUsuarioLogueado].estado=true
    document.querySelector("#msjDisponibilidad").innerHTML="Local habilitado"
}

function deshabilitar(){
    listaLocales[posUsuarioLogueado].estado=false
    document.querySelector("#msjDisponibilidad").innerHTML="Local deshabilitado"
}

function deshabilitarBtnHabilitar(){
    //Si el cupo maximo es menor o igual a la cantidad de cupos ocupados se debe
    //deshabilitar la opcion de volver a habilitar el local
    if(listaLocales[posUsuarioLogueado].cuposOcupados>=listaLocales[posUsuarioLogueado].cupoMaximo){
        document.querySelector("#btnHabilitar").disabled=true
        listaLocales[posUsuarioLogueado].estado=false
        document.querySelector("#msjDisponibilidad").innerHTML="Local deshabilitado"
    }
}

// ----- Cambiar las reservas de estado -----
-----

function mnuEstadoReservas(){
    ocultarFuncionLocal();
    document.querySelector("#mnuEstadoReservas").style.display="block";
    cargarSelectReservasPendientes();
}

function cargarSelectReservasPendientes(){
    document.querySelector("#sctReservaACambiar").innerHTML = "";
}

```

```

let options = "";

//Primero recorrer los objetos persona
for(let pos=0;pos<listaPersonas.length;pos++){
    //Luego recorrer los objetos reserva de la persona
    for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
        let objReserva=listaPersonas[pos].listaReservas[pos1]
        //Si se encuentran reserva pendientes con el mismo nombre que el
Local logueado que las muestre.
        //Tiene que coincidir con el nombre del local logueado y ademas debe
tener estado pendiente.
        if(listaLocales[posUsuarioLogueado].nombre==objReserva.localElegido
&& objReserva.estadoReserva=="pendiente"){
            //Se muestra el usuario del que reservo y la cantidad que reservo
            options+=""

        }
    }
}

document.querySelector("#sctReservaACambiar").innerHTML = options;
}

//esta funcion es igual a la de cargar el select pero para mostrar en el select
antes se fija si coincide con lo que busco el usuario
function buscarNombre(){
    let nombreABuscar=document.querySelector("#txtBuscarNombre").value
    document.querySelector("#msjBusqueda").innerHTML = ""
    let options = "";

    for(let pos=0;pos<listaPersonas.length;pos++){
        for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
            let objReserva=listaPersonas[pos].listaReservas[pos1]
            if(listaLocales[posUsuarioLogueado].nombre==objReserva.localElegido
&& objReserva.estadoReserva=="pendiente"){
                if(listaPersonas[pos].nombre.includes(nombreABuscar)){
                    options+=""
                }
            }
        }
    }
}

```

```

    }
    }
}
document.querySelector("#sctReservaACambiar").innerHTML = options;

if(options==""){
    document.querySelector("#msjBusqueda").innerHTML = "No hay clientes con ese
nombre que tengan una reserva pendiente"
}
}

function finalizarReserva(){
    let idPersonaQueReservo=document.querySelector("#sctReservaACambiar").value;

    //Primero recorrer los objetos persona
    for(let pos=0;pos<listaPersonas.length;pos++){
        //con este si encuentro a la persona con id que hizo la reserva
        if(listaPersonas[pos].id==idPersonaQueReservo){
            //Recorro las reservas de esa persona hasta encontrar la que coincida
            con el local logueado
            for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
                let objReserva=listaPersonas[pos].listaReservas[pos1]
                //Cuando la encuentra le cambio la propiedad estadoReserva del
                objeto reserva a "finalizada"
                // la segunda parte del && es para que no me cambie las
                canceladas
                if(objReserva.localElegido==listaLocales[posUsuarioLogueado].nomb
re && objReserva.estadoReserva=="pendiente"){
                    objReserva.estadoReserva="finalizada"
                    document.querySelector("#msjEstadoReservas").innerHTML = "Se
ha finalizado la reserva"

                }
            }
        }
    }
    cargarSelectReservasPendientes();//Se vuelve a cargar el select tomando en
cuenta que una dejo de ser "pendiente"
}

// ----- Cambiar el cupo maximo -----

```

```

function mnuCupoMaximo(){
    ocultarFuncionLocal();
    document.querySelector("#mnuCupoMaximo").style.display="block";

    //Se muestra el cupo maximo actual del local
    document.querySelector("#msjCupoMaximoActual").innerHTML="El cupo maximo
actual es de "+listaLocales[posUsuarioLogueado].cupoMaximo+" lugares";
}

function cambiarCupoMaximo(){
    let
nuevoCupoMaximo=Number(document.querySelector("#txtCupoMaximoNuevo").value);

    if(noHayReservasPendientes()){
        if(nuevoCupoMaximo>0){//El cupo maximo nuevo debe ser mayor que 0
            listaLocales[posUsuarioLogueado].cupoMaximo=nuevoCupoMaximo;
            mnuCupoMaximo();
            document.querySelector("#msjCupoMaximo").innerHTML="Se ha modificado el
cupo maximo"
        }
        else{
            document.querySelector("#msjCupoMaximo").innerHTML="Digitar un numero
mayor que 0"
        }
    }
    else{
        document.querySelector("#msjCupoMaximo").innerHTML="Hay una reserva
pendiente, no se puede modificar el cupo maximo"
    }
    document.querySelector("#txtCupoMaximoNuevo").value=""
}

function noHayReservasPendientes(){
    let resultado=true;

    //Primero recorrer los objetos persona
    for(let pos=0;pos<listaPersonas.length && resultado;pos++){
        //Luego recorrer los objetos reserva de la persona
        for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length &&
resultado;pos1++){
            let objReserva=listaPersonas[pos].listaReservas[pos1];
            //Si el la reserva esta hecha en el local logueado y esta en estado
pendiente esta funcion retorna false, es decir que si hay reservas pendientes
            if(objReserva.localElegido==listaLocales[posUsuarioLogueado].nombre){

```

```

        if(objReserva.estadoReserva=="pendiente"){
            resultado=false;
        }
    }
}
return resultado
}

// ----- Ver informacion estadistica -----
----

function mnuInfoEstadisticaLocales(){
    ocultarFuncionLocal();
    document.querySelector("#mnuInfoEstadisticaLocales").style.display="block";
    promedioCalificacionLocalLogueado();//Esta funcion muestra solamente el
promedio de calificaciones del local logueado
    promedioCalificacionLocales();//Esta funcion muestra la tabla con todas los
promedios de calificaciones
    let qPendientes=buscarReservasPendientesDelLocal()
    let qFinalizadas=buscarReservasFinalizadasDelLocal()

    let cupoMaximo=listaLocales[posUsuarioLogueado].cupoMaximo

    let porcentajeOcupacion=(qPendientes/cupoMaximo)*100
    document.querySelector("#porcentajeOcupacion").innerHTML="Porcentaje de
ocupacion: "+porcentajeOcupacion+"%"

    document.querySelector("#reservasPendientesLocal").innerHTML="Reservas
pendientes: "+qPendientes
    document.querySelector("#reservasFinalizadasLocal").innerHTML="Reservas
finalizadas: "+qFinalizadas
}

function promedioCalificacionLocalLogueado(){
    //Esta funcion muestra solamente el promedio de calificaciones del local
logueado
    let suma=0
    let cantidad=listaLocales[posUsuarioLogueado].calificaciones.length
    let promedio=0

    for(pos=0;pos<cantidad;pos++){

```

```

        let calificacion=listaLocales[posUsuarioLogueado].calificaciones[pos]
        suma=suma+calificacion
    }
    promedio=suma/cantidad

    if(cantidad>0){
        document.querySelector("#promedioCalificacion").innerHTML="Promedio de
calificaciones: "+promedio
    }
    else{
        document.querySelector("#promedioCalificacion").innerHTML="Promedio de
calificaciones: "+ "Aun no hay calificaciones"
    }
}

function promedioCalificacionLocales(){
    document.querySelector("#tablaCalificaciones").innerHTML="";
    let tablaHTML = "<table>"
    tablaHTML += "<tr><th>Local</th><th>Promedio de Calificaciones</th></tr>"

    for(let pos=0;pos<listaLocales.length;pos++){
        let sumaCalificaciones=0
        //la cantidad de calificaciones que hay es el largo de la lista de
calificaciones que tiene el local
        let cantidadCalificaciones=listaLocales[pos].calificaciones.length

        //este for recorre la lista de calificaciones de cada local y suma cada
una a la variable sumaCalificaciones
        for(let pos1=0;pos1<listaLocales[pos].calificaciones.length;pos1++){
            sumaCalificaciones+=listaLocales[pos].calificaciones[pos1];
        }

        let promedioCalificacion=sumaCalificaciones/cantidadCalificaciones

        tablaHTML+="<tr><td>"+listaLocales[pos].nombre+"</td>";
        if(cantidadCalificaciones>0){
            tablaHTML+="<td>"+promedioCalificacion+"</td>";
        }
        else{
            tablaHTML+="<td>"+ "No hay calificaciones"+"</td>";
        }
    }

    document.querySelector("#tablaCalificaciones").innerHTML =tablaHTML;

```

```

}

function buscarReservasPendientesDelLocal(){
    let cantidadReservasPendientesDelLocal=0
    //Recorrer la lista de personas
    for(let pos=0;pos<listaPersonas.length;pos++){
        //Recorrer la lista de reservas de cada persona
        for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
            let objReserva=listaPersonas[pos].listaReservas[pos1]
            //Buscar una reserva que este en estado pendiente y que coincida con
            el nombre del local logueado
            if(objReserva.estadoReserva=="pendiente" &&
objReserva.localElegido==listaLocales[posUsuarioLogueado].nombre){
                //si encuentra le suma la cantidad de reservas en estado
                pendientes que tiene esa persona para ese local
                cantidadReservasPendientesDelLocal=cantidadReservasPendientesDe
ocal+objReserva.cantidadReservada
            }
        }
    }
    return cantidadReservasPendientesDelLocal
}

//Esta funcion es igual a la anterior pero busca la cantidad de reservas en
estado "finalizada"
function buscarReservasFinalizadasDelLocal(){
    let cantidadReservasFinalizadasDelLocal=0
    //Recorrer la lista de personas
    for(let pos=0;pos<listaPersonas.length;pos++){
        //Recorrer la lista de reservas de cada persona
        for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
            let objReserva=listaPersonas[pos].listaReservas[pos1]
            //Buscar una reserva que este en estado pendiente y que coincida con
            el nombre del local logueado
            if(objReserva.estadoReserva=="finalizada" &&
objReserva.localElegido==listaLocales[posUsuarioLogueado].nombre){
                //si encuentra le suma la cantidad de reservas en estado
                pendientes que tiene esa persona para ese local
                cantidadReservasFinalizadasDelLocal=cantidadReservasFinalizadasDe
lLocal+objReserva.cantidadReservada
            }
        }
    }
    return cantidadReservasFinalizadasDelLocal
}

```

```

// ----- FUNCIONES PERSONA -----
// ----- Solicitar reserva -----

let listaReservas=[]; //Lista global que va a tener los objetos reserva

function mnuSolicitarReserva(){
    ocultarFuncionPersona();
    document.querySelector("#mnuSolicitarReserva").style.display="block";
    CargarSelectLocales();
    cargarSelectLocalesPendientes();
}

function CargarSelectLocales(){
    // Limpiar el select
    document.querySelector("#sctLocalAReservar").innerHTML = "";
    // Recorrer la lista de locales
    let options = "";
    for (let objLocal of listaLocales){
        // Agrega un local al select
        if(objLocal.estado){ //Este if causa que solo se muestren en el select los
locales habilitados
            options+="

```



```

        if(cantidadElegida>0){//Se debe seleccionar un numero mayor a
0
            if(cantidadElegida<=(objLocal.cupoMaximo-
objLocal.cuposOcupados)){

                objLocal.cuposOcupados=objLocal.cuposOcupados+cantida
dElegida

                document.querySelector("#msjReserva").innerHTML="Se
han solicitado "+cantidadElegida+" cupos"
                document.querySelector("#txtCantidadAReservar").value
=""
                document.querySelector("#sctLocalAReservar").value=""

                let objReserva=new
reserva(localElegido,cantidadElegida,"pendiente");//Se crea un objeto reserva
                listaPersonas[posUsuarioLogueado].listaReservas.push(
objReserva);//Se guarda el objeto reserva en la lista de reservas

                cargarSelectLocalesPendientes();
            }
            else{
                document.querySelector("#msjReserva").innerHTML="No
hay suficientes cupos disponibles"
            }
        }
        else{
            document.querySelector("#msjReserva").innerHTML="Digitar
un numero mayor a 0"
        }
    }
    else{
        document.querySelector("#msjReserva").innerHTML="Ya tiene una
reserva pendiente en ese local"
    }
}

    if(objLocal.cupoMaximo==objLocal.cuposOcupados){//Si el total de cupos
ocupados alcanza el cupo maximo se deshabilita el local
        objLocal.estado=false
    }
}
}
}

```

```

//Esta funcion se fija si dentro de el objeto persona que se logueo existe alguna
reserva en el local el cual esta intentando reservar
//Si no hay reservas pendientes en ese local retorna true
//En caso opuesto retorna false
function reservaPendiente(localElegido){
    let NohayReservasPendientes=true;

    for(let pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length &&
NohayReservasPendientes;pos++){
        let
reservaPendiente=listaPersonas[posUsuarioLogueado].listaReservas[pos].localElegido
o
        let
estadoReserva=listaPersonas[posUsuarioLogueado].listaReservas[pos].estadoReserva

        //para que se considere que hay una reserva pendiente debe coincidir el
local que eligio el usuario para reservar
        //con el nombre en el listado de sus reservas pendientes
        //y esa reserva debe tener estado pendiente
        if(reservaPendiente==localElegido && estadoReserva=="pendiente"){

            NohayReservasPendientes=false;
        }
    }

    return NohayReservasPendientes
}

// ----- Cancelar reserva -----

function cargarSelectLocalesPendientes(){

    let options="";

    //Este for recorre la lista de reservas pendientes que tiene el usuario que
se logueo y las agrega al select de reservas pendientes
    for(let
pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        if(listaPersonas[posUsuarioLogueado].listaReservas[pos].estadoReserva=="p
endiente"){//Se agregan al select las que tienen estado pendiente

```

```

        let
        reservaPendientesDelUsuarioLogueado=listaPersonas[posUsuarioLogueado].listaReservas[pos].localElegido;

        options+="<option
value='"+reservaPendientesDelUsuarioLogueado+"'>"+reservaPendientesDelUsuarioLogueado+"</option>"
    }
}

document.querySelector("#sctLocalACancelar").innerHTML=options;
}

function cancelarReserva(){

    let reservaACancelar=document.querySelector("#sctLocalACancelar").value

    //este for recorre la lista de reservas que tiene el usuario logueado hasta encontrar la que coincide con
    //la que el usuario quiere cancelar. Cuando la encuentra le cambia el estado(atributo) a "cancelada"
    for(let
pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        let
reservaPendienteDelUsuarioLogueado=listaPersonas[posUsuarioLogueado].listaReservas[pos].localElegido;
        if(reservaACancelar==reservaPendienteDelUsuarioLogueado){

            //En este for se recorre la lista de locales hasta encontrar la que el usuario cancelo
            //cuando la encuentra le resta a su atributo "cuposOcupados" la cantidad que habia
            //reservado el usuario.
            for(let pos1=0;pos1<listaLocales.length;pos1++){
                if(listaLocales[pos1].nombre==reservaACancelar &&
listaPersonas[posUsuarioLogueado].listaReservas[pos].estadoReserva=="pendiente"){

                    listaLocales[pos1].cuposOcupados=listaLocales[pos1].cuposOcupados-listaPersonas[posUsuarioLogueado].listaReservas[pos].cantidadReservada;

                    //Luego de la resta anterior se le cambia el estado a la reserva a "cancelada".

```

```

        listaPersonas[posUsuarioLogueado].listaReservas[pos].estadoReserva="cancelada"
    }
}

}

document.querySelector("#msjCancelarReserva").innerHTML="Se ha cancelado la reserva"

cargarSelectLocalesPendientes();//Vuelve a mostrar el select con las que siguen pendientes
}

// ----- Calificar reserva -----

function mnuCalificarReserva(){
    ocultarFuncionPersona();
    document.querySelector("#mnuCalificarReserva").style.display="block";
    cargarSelectReservasFinalizadas();
}

function cargarSelectReservasFinalizadas(){

    let options="";

    //Este for recorre la lista de reservas finalizadas que tiene el usuario que se logueo y las agrega al select de reservas finalizadas
    for(let
pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        let objReserva=listaPersonas[posUsuarioLogueado].listaReservas[pos]
        if(objReserva.estadoReserva=="finalizada" && objReserva.noCalifico){//Se agregan al select las que tienen estado finalizada y que no han sido calificadas
            let
reservaPendientesDelUsuarioLogueado=listaPersonas[posUsuarioLogueado].listaReservas[pos].localElegido;

            options+="

```

```

        document.querySelector("#sctLocalACalificar").innerHTML=options;
    }

function calificarReservaFinalizada(){
    let unaCalificacion=Number(document.querySelector("#txtCalificacion").value);
    let localElegido=document.querySelector("#sctLocalACalificar").value;

    if(unaCalificacion>0 && unaCalificacion<=5){
        let posObjLocal = buscarLocal(localElegido);
        let objLocal=listaLocales[posObjLocal];

        objLocal.agregarCalificacion(unaCalificacion);
        document.querySelector("#msjAgregarCalificacion").innerHTML =
"Calificacion agregada correctamente";
        document.querySelector("#txtCalificacion").value = "";

        //Luego de calificar se cambia la propiedad "noCalifico" de la reserva a
false
        for(pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos+
+){
            if(listaPersonas[posUsuarioLogueado].listaReservas[pos].localElegido=
=localElegido){
                listaPersonas[posUsuarioLogueado].listaReservas[pos].noCalifico=f
alse;
            }
        }

        cargarSelectReservasFinalizadas();//Si logra calificar se vuelve a cargar
el select

    }
    else{
        document.querySelector("#msjAgregarCalificacion").innerHTML = "Digitar
una calificacion valida";
    }
}

function buscarLocal(localElegido){
    let resultado=-1

    for (let pos = 0; pos < listaLocales.length && resultado===-1 ; pos++) {
        let objLocal = listaLocales[pos];
    }
}

```

```

        if(objLocal.nombre==localElegido){
            resultado=pos;
        }
    }
    return resultado
}

// ----- Mostrar reservas pendientes -----
-----

function mnuReservasPendientes(){
    ocultarFuncionPersona();

    let htmlReservasPendientes="";

    for (let
pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        let objReserva=listaPersonas[posUsuarioLogueado].listaReservas[pos]

        if(objReserva.estadoReserva=="pendiente"){
            let posObjLocal=-1;

            //Con este for recorro la lista de locales y encuentro la posicion
del Local el cual la persona tiene la reserva pendiente
            for(let pos1=0;pos1<listaLocales.length;pos1++){
                let objLocal=listaLocales[pos1]
                if(objLocal.nombre==objReserva.localElegido){
                    posObjLocal=pos1;
                }
            }

            htmlReservasPendientes += "<div>";
            htmlReservasPendientes += "<h2>"+ listaLocales[posObjLocal].nombre
+"</h2>";
            htmlReservasPendientes += "<img
src='"+listaLocales[posObjLocal].foto+"'>";
            htmlReservasPendientes += "<p>"+"Cantidad de cupos reservados: "+
objReserva.cantidadReservada +"</p>";
            htmlReservasPendientes += "</div>";
            htmlReservasPendientes += "<hr>"

        }
    }
}

```

```

    document.querySelector("#lstReservasPendientes").innerHTML=htmlReservasPendie
ntes;
    document.querySelector("#mnuReservasPendientes").style.display="block";

    if(htmlReservasPendientes==""){//Si no hay reservas pendientes mostrar un
mensaje
    document.querySelector("#msjReservasPendientes").innerHTML="No hay reservas
pendientes"
    }
}

// ----- Info estadistica personas -----
--

function mnuInfoEstadisticaPersonas(){
    ocultarFuncionPersona();
    document.querySelector("#mnuInfoEstadisticaPersonas").style.display="block";
    let tablaHTML = "<table>"
    tablaHTML += "<tr><th>Reservas finalizadas</th><th>Reservas
completadas</th><th>Reservas del local</th><th>Q reservada/Reservas del
local</th></tr>"

    //Se recorren las reservas del usuario logueado
    for(pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        let objReserva=listaPersonas[posUsuarioLogueado].listaReservas[pos]

        //Si la reserva esta finalizada se muestra lo siguiente
        if(objReserva.estadoReserva=="finalizada"){
            tablaHTML += "<tr><td>" +objReserva.localElegido+"</td>";
            tablaHTML += "<td>" +objReserva.cantidadReservada+"</td>";

            let nombreLocal=objReserva.localElegido
            let cantidadReservadaTotal=cantidadReservada(nombreLocal)

            tablaHTML += "<td>" +cantidadReservadaTotal+"</td>";
            tablaHTML +=
"<td>" +(objReserva.cantidadReservada/cantidadReservadaTotal)*100+"%"+"</td>";

        }

    }

    localConMasReservas();

```

```

        document.querySelector("#tablaReservas").innerHTML=tablaHTML;
    }

function cantidadReservada(nombreLocal){
    //Esta funcion recorre todas las reservas que han hecho los usuarios y cuenta
    la cantidad total segun el nombre del local
    let cantidadReservadaTotal=0;

    for(let pos=0;pos<listaPersonas.length;pos++){
        for(let pos1=0;pos1<listaPersonas[pos].listaReservas.length;pos1++){
            let objReserva=listaPersonas[pos].listaReservas[pos1]
            if(objReserva.localElegido==nombreLocal){
                cantidadReservadaTotal=cantidadReservadaTotal+objReserva.cantidad
Reservada
            }
        }
    }
    return cantidadReservadaTotal
}

function localConMasReservas(){

    let localMasReservado=""
    let mayorQDeReservas=0
    //Recorro la lista de reservas del usuario logueado
    for(let
pos=0;pos<listaPersonas[posUsuarioLogueado].listaReservas.length;pos++){
        let objReserva=listaPersonas[posUsuarioLogueado].listaReservas[pos]
        if(mayorQDeReservas<objReserva.cantidadReservada){
            mayorQDeReservas=objReserva.cantidadReservada
        }
        //Si hay mas de un local que coincide en el maximo se muestran todos
        if(mayorQDeReservas==objReserva.cantidadReservada){
            localMasReservado+=objReserva.localElegido+", "
        }
    }

    document.querySelector("#mayorQReservas").innerHTML="Local/es mas
reservado/s: "+localMasReservado+" con "+mayorQDeReservas+" reservas";
}

```


JavaScript – class.js

```
let ultimoIdLocal = 0;
class local{
    constructor(pNombre,pUsuario,pContraseña,pTipoLocal,pDireccion,pCupoMaximo,pFoto,pEstado,pCuposOcupados){
        ultimoIdLocal++;
        this.id=ultimoIdLocal;
        this.nombre=pNombre;
        this.usuario=pUsuario;
        this.contraseña=pContraseña;
        this.tipoLocal=pTipoLocal;
        this.direccion=pDireccion;
        this.cupoMaximo=pCupoMaximo;
        this.foto=pFoto;
        this.estado=pEstado;
        this.cuposOcupados=pCuposOcupados;
        this.calificaciones = [];

    }

    agregarCalificacion(valor){
        this.calificaciones.push(valor);
    }
}

let ultimoIdPersona = 0;
class persona{
    constructor(pNombre,pUsuario,pContraseña){
        ultimoIdPersona++;
        this.id=ultimoIdPersona;
        this.nombre=pNombre;
        this.usuario=pUsuario;
        this.contraseña=pContraseña;
        this.listaReservas=[];
    }
}

class reserva{
    constructor(pLocalElegido,pCantidadReservada,pEstadoReserva){
        this.localElegido=pLocalElegido;
        this.cantidadReservada=pCantidadReservada;
        this.estadoReserva=pEstadoReserva;
    }
}
```

```
        this.noCalifico=true;
    }
}
```

INFORMACION PRECARGADA

Usuarios Local:

```
0: local
  ▶ calificaciones: []
  contraseña: "123"
  cupoMaximo: 5
  cuposOcupados: 0
  direccion: "Av wilson 12"
  estado: true
  foto: "fotos/foto4.jpg"
  id: 1
  nombre: "Francis"
  tipoLocal: "restaurant"
  usuario: "rest1"
```

```
▼ 2: local
  ▶ calificaciones: []
  contraseña: "323"
  cupoMaximo: 10
  cuposOcupados: 0
  direccion: "Maximo Tajes 56"
  estado: true
  foto: "fotos/foto6.jpg"
  id: 3
  nombre: "Gourmet"
  tipoLocal: "restaurant"
  usuario: "rest3"
  ▶ [[Prototype]]: Object
```

```
▼ 4: local
  ▶ calificaciones: []
  contraseña: "123"
  cupoMaximo: 8
  cuposOcupados: 0
  direccion: "Valdivia 16"
  estado: true
  foto: "fotos/foto2.jpg"
  id: 5
  nombre: "Mucha Arte"
  tipoLocal: "museo"
  usuario: "mus2"
  ▶ [[Prototype]]: Object
```

```
▼ 6: local
  ▶ calificaciones: []
  contraseña: "122"
  cupoMaximo: 4
  cuposOcupados: 0
  direccion: "Portezuelo 8"
  estado: true
  foto: "fotos/foto3.jpg"
  id: 7
  nombre: "Grecia"
  tipoLocal: "teatro"
  usuario: "tea2"
  ▶ [[Prototype]]: Object
```

```
▼ 1: local
  ▶ calificaciones: []
  contraseña: "223"
  cupoMaximo: 3
  cuposOcupados: 0
  direccion: "Av Italia 720"
  estado: true
  foto: "fotos/foto5.jpg"
  id: 2
  nombre: "Hard Rock"
  tipoLocal: "restaurant"
  usuario: "rest2"
  ▶ [[Prototype]]: Object
```

```
▼ 3: local
  ▶ calificaciones: []
  contraseña: "113"
  cupoMaximo: 2
  cuposOcupados: 0
  direccion: "Av Uruguay 2666"
  estado: true
  foto: "fotos/foto1.jpg"
  id: 4
  nombre: "Louvre"
  tipoLocal: "museo"
  usuario: "mus1"
  ▶ [[Prototype]]: Object
```

```
▼ 5: local
  ▶ calificaciones: []
  contraseña: "121"
  cupoMaximo: 3
  cuposOcupados: 0
  direccion: "Divina Comedia 134"
  estado: true
  foto: "fotos/foto8.jpg"
  id: 6
  nombre: "Shakespeare"
  tipoLocal: "teatro"
  usuario: "tea1"
```

Usuarios Persona:

```
0: persona
  contraseña: "123"
  id: 1
  listaReservas: Array(2)
    0: reserva {localElegido: 'Francis', cantidadReservada: 3, estadoReserva: 'finalizada', noCalifico: true}
    1: reserva {localElegido: 'Grecia', cantidadReservada: 2, estadoReserva: 'pendiente', noCalifico: true}
    length: 2
    [[Prototype]]: Array(0)
  nombre: "Martin"
  usuario: "user1"
  [[Prototype]]: Object
```

```
1: persona
  contraseña: "123"
  id: 2
  listaReservas: Array(2)
    0: reserva {localElegido: 'Hard Rock', cantidadReservada: 4, estadoReserva: 'finalizada', noCalifico: true}
    1: reserva {localElegido: 'Louvre', cantidadReservada: 1, estadoReserva: 'pendiente', noCalifico: true}
    length: 2
    [[Prototype]]: Array(0)
  nombre: "Martina"
  usuario: "user2"
  [[Prototype]]: Object
```

```
2: persona
  contraseña: "123"
  id: 3
  listaReservas: Array(1)
    0: reserva {localElegido: 'Shakespeare', cantidadReservada: 2, estadoReserva: 'pendiente', noCalifico: true}
    length: 1
    [[Prototype]]: Array(0)
  nombre: "Pedro"
  usuario: "user3"
  [[Prototype]]: Object
```

```
3: persona
  contraseña: "123"
  id: 4
  listaReservas: []
  nombre: "Juan"
  usuario: "user4"
  [[Prototype]]: Object
```

```
4: persona
  contraseña: "123"
  id: 5
  listaReservas: []
  nombre: "Santiago"
  usuario: "user5"
  [[Prototype]]: Object
```