|  |
| --- |
|  |
| Universidad ORT Uruguay |
| **TUreservaWEB** |
| **A person smiling for the camera  Description automatically generated with medium confidence**    **Emiliano Luján – 237633**  **A picture containing person, person, sky, outdoor  Description automatically generated**  **Gustavo Brañas – 282436**  **Grupo N1B**  **Docente: Martin Luz**  **Analista en Tecnologías de la Información**  **Junio 2022** |
|  |
|  |
|  |

**Tabla de contenido**

[RESUMEN 3](#_Toc106643841)

[ENUMERACIÓN DE LAS FUNCIONALIDADES DESARROLLADAS. 4](#_Toc106643842)

[VARIABLES GLOBALES 5](#_Toc106643843)

[DESCRIPCIÓN DE LAS FUNCIONES IMPLEMENTADAS 6](#_Toc106643844)

[Operaciones relativas a ambos usuarios 6](#_Toc106643845)

[Funcionalidades del usuario Local 8](#_Toc106643846)

[Funcionalidades del usuario Persona 10](#_Toc106643847)

[DATOS PRECARGADOS 13](#_Toc106643848)

[OPORTUNIDADES DE MEJORA 14](#_Toc106643849)

# RESUMEN

El objetivo de este proyecto es generar una aplicación de uso tanto para locales (restaurantes, teatros y museos) y clientes que podrán administrar reservas teniendo en cuenta el cupo disponible. El cometido principal de la aplicación es que los clientes puedan reservar, calificar, ver puntuaciones y cancelar reservas en diferentes locales y, por otro lado, que los locales sean capaces de permitir o no la realización de reservas, su modificación en base a los cupos disponibles y visualización, junto con un análisis estadístico basado en la calificación de los clientes.

# ENUMERACIÓN DE LAS FUNCIONALIDADES DESARROLLADAS.

Se desarrollaron las siguientes funcionalidades en base a dos tipos de usuarios: Local y Persona.

F01 – Registrarse en la aplicación - Usuario: Persona.

F02 – Ingresar en la aplicación - Usuarios: Local y Persona.

F03 – Visualizar información estadística - Usuarios: Local y Persona.

F04 – Habilitar y Deshabilitar disponibilidad - Usuario: Local.

F05 – Cambiar reservas de estado - Usuario: Local.

F06 – Modificar cupos máximos - Usuario: Local.

F07 – Realizar solicitudes de reserva - Usuario: Persona.

F08 – Cancelar reserva - Usuario: Persona.

F09 – Calificar reserva - Usuario: Persona.

F10 – Ver listado de reservas pendientes - Usuario: Persona.

# VARIABLES GLOBALES

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Estructura de datos empleada** | **Descripción/Observaciones** |
| miUsuario | Array asociativo | Almacena la información del usuario activo |
| usuariosPersona | Array indexado que contiene arrays asociativos. | Almacena la información de los usuarios tipo Persona precargados o registrados mediante la aplicación (cada array asociativo contiene los datos de un usuario). Se detalla la información incluida en la class RegistroPersona. |
| usuariosLocal | Array indexado que contiene arrays asociativos. | Almacena la información de los usuarios tipo Local precargados en la aplicación (cada array asociativo contiene los datos de un usuario). No es posible el registro de usuarios tipo Local mediante esta aplicación. |
| reservas | Array indexado que contiene arrays asociativos. | Almacena la información de las reservas realizadas (cada array asociativo contiene los datos que identifican a una reserva). Se detalla la información incluida en la class Reservas. |

# DESCRIPCIÓN DE LAS FUNCIONES IMPLEMENTADAS

## Operaciones relativas a ambos usuarios

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Ingreso al sistema | Al ingresar, se despliega un menú que contiene en su parte superior, el nombre del usuario activo (usuario que ingresa a la aplicación) y en la inferior una serie de enlaces a los diferentes servicios que brida la aplicación asociados a la función modificarMenu().  A su vez, se despliega una página principal que contiene:   * Si el usuario registrado es perfil *Persona*: la lista de los locales con que la aplicación dispone. * Si el usuario es perfil *Local*: se muestra una estadística (no real y meramente ilustrativa) de como podría llegar a ser el comportamiento de las reservas en el último mes para ese local. |
| Encuéntranos en las redes | Esta funcionalidad permite acceder, mediante evento onclick, a páginas web del tipo Facebook, Instagram y Twitter correspondientes a las redes sociales de la aplicación. (No se cuenta con accesos reales, solo se redirige a las portadas de las redes correspondientes) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FUNCIONES DE ACCESO INDEPENDIENTE AL PERFIL** | | | | |
| **Función principal** | **Parámetros** | **Retorno** | **Funciones invocadas desde la principal** | **Descripción** |
| FUNCIONALIDAD: F02  onClickIngresar()  Evento: onclick  Esta función se llama desde el onLoad de la aplicación. | n/a | Si usuario no registrado, mal ingresado par usuario contraseña o no se ingresan caracteres en las entradas: mensaje de alerta.  Si usuario ingresado correctamente:  La variable global miUsuario almacena los datos del usuario Persona ingresado. | * eliminarEspacios() * validarIngreso() * cambioVisibilidad() * finalizarReservas() * mensajesAlerta() * borrarDatos();   Si PERSONA:   * reservasPendientesPersona() * cancelacionReservas() * calificacionLocales()   Si LOCAL:  - calificacionDeLocales()  - estadisticasLocal() | Permite el ingreso de usuarios registrados a la aplicación. Se cargan nuevas funcionalidades según sea el perfil ingresado. |
| modificarMenu(str)  Evento: onclick | String. En este se detalla el menú que se desea acceder | n/a | n/a | Botón que permite al usuario el acceso a los diferentes menús de la aplicación, cambiando la visibilidad de estos y retornando los valores de input y select a default. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Parámetros** | **Retorno** | **Descripción** |
| eliminarEspacios(str) | String | String en los que se le eliminó los espacios | Eliminación de espacios en el string. |
| validarIngreso(str1,str2) | 2 strings: usuario y contraseña ingresado por usuario | Modificación de la variable miUsuario.  Si no se encuentra coincidencia, función retorna *null*. | Permite la búsqueda de usuarios del tipo Local o Persona chequeando en la base de datos si existe coincidencia con los argumentos ingresados. |
| cambioVisibilidad(str1,str2,str3) | 3 Strings.  Tipo (. o #), string correspondiente al identificador, habilitación (block, none) | n/a | Permite ocultar o mostrar determinada sección según se solicite |
| mensajesAlerta(str1,str2) | 2 Strings.  String(mensaje) y string tipo de alerta | n/a | Despliega mensajes del tipo Bienvenida, Error, Aceptación o Cancelación de acciones, Alerta o Mensajes de aviso que no requieren interacción con el usuario. |
| borrarDatos() | n/a | n/a | Permite borrar los datos de inputs en las contraseñas en el menú de ingreso o de registro a la aplicación |
| onClickSalir() | n/a | n/a | Cambia la visibilidad de los diferentes menús y retorna a la página de ingreso. Muestra un mensaje de despedida al usuario. |

## Funcionalidades del usuario Local

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funcionalidad** | **Función** | **Parámetros** | **Retorno** | **Descripción/Observaciones** |
| F03 | estadisticasLocal() | n/a | n/a | Establece las variables que indican la posición dentro del HTML donde se imprimirán las estadísticas del local y las estadísticas comparativas con otros locales.  Esta función llama a dos funciones más:  estadisticas(strHTML)  estadisticasComparativas(strHTML) |
| estadisticas(strHTML) | strHTML corresponde a la ubicación en el HTML donde se imprimirá la tabla estadística del local. | n/a | Calcula el porcentaje de ocupación e imprime en pantalla las estadísticas del local.  Esta función llama a dos funciones más:  asignacionEstrellas(str)  calificacionDeLocales(str, true) |
| asignacionEstrellas(str) | Como parámetro considera la calificación del local obtenida desde el array miUsuario | Imagen | Convierte la puntuación numérica del local a una imagen de estrellas que se imprimirá en la pantalla. |
| calificacionDeLocales(str, true) | Recibe el parámetro de id del local y true. | Si true se retorna la cantidad de reservas finalizadas para este local. | Calcula la calificación de las reservas Finalizadas que hayan sido calificadas previamente por usuarios y modifica el key calificación de miUsuario, correspondiente al Local. |
| estadisticasComparativas(strHTML) | strHTML corresponde a la ubicación en el HTML donde se imprimirá la tabla estadística comparativa entre locales. | n/a | Calcula la calificación de todos los locales e imprime en pantalla las estadísticas del local en comparación con los demás.  Esta función llama a la función:  asignacionEstrellas(str) |
| F04 | modificarDisponibilidad() | n/a | n/a | Modifica el key habilitado del usuario Local a true o false siempre que no existan reservas en estado Pendiente. |
| F05 | buscarPersona() | n/a | n/a | Permite la búsqueda de usuarios con reservas en el local activo. La búsqueda es amplia: si no se encuentra usuario o no se ingresa el mismo se muestra un mensaje de error. |
| coincidenciasPersona() | String | Array de usuarios con reserva Pendiente en ese local. De lo contrario, retorna ( -1). | Permite la búsqueda de usuarios según los caracteres ingresados en el string con reservas Pendientes para el local activo. Si retorna (-1) se muestra un mensaje en pantalla de error en la búsqueda. |
| finalizarReservas(array) | Opcional: Array.  Se emplea en la función buscarPersona() y considera como parámetro para esta función el retorno de coincidenciasPersona() | n/a | Permite el cambio de visibilidad de menus, vacía el contenido de impresiones previas de tabla de reservas e inputs de búsqueda.  Esta función llama a dos funciones más:  reservasPendientes(strHTML, array)  onClickCambioEstado() |
| reservasPendientes(strHTML, array) | -strHTML corresponde a la ubicación en el HTML donde se imprimirá la tabla de posibles reservas a Finalizar.  -array: es el parámetro que toma la función finalizarReservas(). | n/a | Permite la impresión en pantalla de las reservas en estado Pendiente o aquellas incluidas en el array para el local activo, y asocia un botón para poder Finalizarlas. |
| onClickCambioEstado() | n/a | n/a | Modifica la reserva de Pendiente a Finalizado. Aumenta los cuposDisponibles considerando los cupos Pendientes que cambiaron de estado para el local activo. |
| F06 | modificacionCupos() | n/a | n/a | Valida si el ingreso de cupos corresponde a un entero positivo. Si no es así, muestra una alerta. Corrobora si existen reservas Pendientes para el local y de ser así, alerta al usuario y no se modifican los cupos máximos. De lo contrario se modifican los cupos disponibles y máximo considerando este nuevo valor. Si las reservas se encuentran Deshabilitadas, se alerta al usuario luego de cambiar el cupo máximo. |

## Funcionalidades del usuario Persona

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funcionalidad** | **Función** | **Parámetros** | **Retorno** | **Descripción/Observaciones** |
| F01 | onClickHablitacionRegistro()  Evento: onclick.  Esta función se llama desde el onLoad de la aplicación. | n/a | n/a | Modifica la visibilidad de los diferentes menús permitiendo que se acceda al Registro de nuevos usuarios. |
| onClickRegistrar()  Evento: onclick.  Esta función se llama desde el onLoad de la aplicación. | n/a | n/a | Si usuario registrado incorrectamente: mensaje de alerta que varía según los errores registrados.  Si usuario registrado correctamente:  Se realiza un push de los datos del usuario a la variable global usuariosPersona. |
| retornoAingreso()  Función asociada directamente al HTML como onclick | n/a | n/a | Permite al usuario que por error ingresó a registro, retornar al menú de ingreso. |
| validacionIngresoNombreApellido(str) | String | Booleano | Permite determinar si se incluye al menos un espacio entre dos textos, identificando de esta forma que se ingresó al menos un nombre y apellido. Si se cumple la condición se evalúa el nombre de usuario ingresado. |
| ValidacionUserName(str)\*\* | String | String (faltaIngreso ó poseeEspacios ó existeNombreDeUsuario) | Función implicada en la validación del nombre de usuario. Despliega mensajes si el dato de nombre de usuario no es ingresado, posee espacio o el nombre de usuario ya existe en la base de datos. Si no retorna valor entonces se evalúa el cumplimiento de las condiciones de la contraseña. |
| validacionContraseña(str)  mediante un onkeyup | String | Booleano | Permite validar la contraseña ingresada cumple con los requisitos solicitados. |
| removerValidacionContraseña() | n/a | n/a | Permite modificar las condiciones de la contraseña desplegada en el HTML a default (todas en color rojo) y devuelve al estado default de los inputs de registro |
| asignacionId() | n/a | String correspondiente a un número | Determina el id del nuevo usuario a registrar considerando el valor máximo de id existente en la variable usuariosPersona. |
| F03 | estadisticaPersona() | n/a | n/a | Establece las variables que indican la posición dentro del HTML donde se imprimirán las estadísticas de la persona y el local con mayores reservas realizadas por el usuario.  Esta función llama a una función más:  tablaEstadisticaPersona(strHTML1, strHTML2); |
| tablaEstadisticaPersona(strHTML1, strHTML2); | strHTML1 y strHTML2 corresponden a la ubicación en el HTML donde se imprimirán la tabla estadística de la Persona junto con el local / locales con mayores reservas realizadas por el usuario respectivamente. | n/a | - Determina la cantidad de reservas de miUsuario en los diferentes estados. Determina el porcentaje de reservas asociado a miUsuario para cada local e imprime en pantalla las estadísticas de la Persona.  - A su vez determina cual es el local (es) con mayor número de reservas e imprime en pantalla el o los nombres de los locales junto con el ícono que representa el rubro.  Si el usuario no posee reservas realizadas, se muestra un mensaje mediante cambioVisibilidad(). |
| F07 | mostrarPorRubro() | n/a | n/a | Genera select con los locales disponibles para reserva (cupos mayores a 0, local habilitado y no presencia de reservas en estado pendiente par miUsuario) e imprime logo del rubro seleccionado. Cambia visibilidad de menú según select. Muestra mensajes de alerta o notificación según selección. |
| infoLocalAreservar() | n/a | n/a | Imprime en pantalla los datos del local a reservar y genera botón para confirmación de reserva. |
| confirmarSeleccionCupos() | n/a | n/a | Pushea los datos de la reserva al array reservas mediante la instancia del constructor new Reservas. Despliega mensajes de confirmación o de error. Modifica la visibilidad de menús.  Actualiza la información luego de estos cambios:  cancelacionReservas()  reservasPendientesPersona()  mostrarPorRubro()  reservasPendientesPersona() |
| F08 | cancelacionReservas() | n/a | n/a | Función que define las variables donde se imprimirán en el HTML las reservas posibles de ser canceladas y a su vez, crea la funcionalidad de los botones de cancelación mediante un evento tipo click en onClickCancelar().  Esta función llama a una función más:  tablaCancelacionReservas(strHTML1, strHTML2, strHTML3); |
| tablaCancelacionReservas(strHTML1, strHTML2, strHTML3) | strHTML1, strHTML2 y strHTML3 corresponden a la ubicación en el HTML donde se imprimirán la tabla de cancelación según se trate del rubro restaurant, teatro o museo respectivamente en base a las reservas Pendientes. | n/a | Realiza cambios de visibilidad según existan reservas Pendientes de miUsuario para cada rubro e imprime junto con el ícono del rubro, la lista de reservas en el HTML y los botones para la cancelación de estas. De no existir reservas, se muestra un mensaje impreso en el HTML. |
| onClickCancelar() | n/a | n/a | Permite el cambio de estado Pendiente a Cancelado, consultado previamente al usuario si realmente quiere ese cambio. Despliega a su vez mensajes según sean las decisiones del usuario. Si ocurre el cambio, modifica los cuposDisponibles del local en cuestión y actualiza datos llamando a:  cancelacionReservas()  reservasPendientesPersona()  estadisticaPersona() |
| F09 | calificacionLocales() | n/a | n/a | Función que define las variables donde se imprimirán en el HTML los locales posibles de ser calificados y a su vez, crea la funcionalidad de los botones de calificación mediante un evento tipo click, en onClickCalificar().  La calificación se realiza mediante un select.  Esta función llama a una función más:  tablaCalificacionLocales (strHTML1) |
| tablaCalificacionLocales (strHTML1) | strHTML1 corresponde a la ubicación en el HTML donde se imprimirá la tabla que permitirá la calificación de los locales. | n/a | Realiza cambios de visibilidad según existan locales a ser calificados por miUsuario e imprime la lista de locales en el HTML, los select para asignar puntaje y los botones para confirmar la calificación. De no existir locales a calificar, se muestra un mensaje impreso en el HTML. |
| onClickCalificar() |  |  | Permite la calificación de los locales, modificando el valor del key calificación dentro de la reserva. Despliega mensajes en caso de no seleccionar opciones o en caso de calificar.  Si ocurre la calificación, actualiza datos llamando a:  calificacionLocales() |
| F10 | reservasPendientesPersona() | n/a | n/a | Esta función solo aporta las variables que proporcionan la ubicación en el HTML donde se imprimirán las reservas de miUsuario desplegadas según rubro de locales.  A pesar de su poca funcionalidad, esta función se crea para mantener una coherencia del código.  Esta función llama a una función más:  tablaReservasPendientes(strHTML1, strHTML2, strHTML3); |
| tablaReservasPendientes(strHTML1, strHTML2, strHTML3) | strHTML1, strHTML2 y strHTML3 corresponden a la ubicación en el HTML donde se imprimirán la tabla de reservas según se trate del rubro restaurant, teatro o museo respectivamente. | n/a | Realiza cambios de visibilidad según existan reservas Pendientes de miUsuario para cada rubro e imprime junto con el ícono del rubro, la lista de reservas en el HTML. De no existir reservas, se muestra un mensaje impreso en el HTML. |

# DATOS PRECARGADOS

|  |  |  |  |
| --- | --- | --- | --- |
| **Usuario** | | | |
| **Persona** | | **Local** | |
| **Nombre** | **Reservas** | **Rubro** | **Nombre** |
| Gustavo Brañas | 3 reservas FINALIZADAS   * Lo de pepe (Calificado) * Garage Burguer (sin Calificar) * El tinglado (sin Calificar)   1 reserva PENDIENTE   * Louvre   1 reserva CANCELADA   * Lo de Pepe | Restaurant | “Lo de Pepe” |
| Emiliano Lujan | 2 reservas PENDIENTES   * Lo de pepe * Garage Burguer | “Garage Burguer” |
| José Diaz | 3 reservas FINALIZADAS   * Lo de pepe (Calificado) * Louvre (Calificado) * Lisa (sin Calificar)   2 reservas PENDIENTES   * Lo de Pepe * Louvre | “Lisa” |
| Joaco Kam | 1 reserva FINALIZADA   * El Galpón (Calificado)   1 reserva PENDIENTE   * Museo del Prado   1 reserva CANCELADA   * Museo del Prado | Museo | "Museo Del Prado" |
| Mailen Arleo | 1 reserva FINALIZADA   * Louvre (Calificado)   1 reserva PENDIENTE   * Lo de Pepe   1 reserva CANCELADA   * Lo de Pepe | "Louvre" |
|  | | Teatro | “El Galpón” |
| "El Tinglado" |

# OPORTUNIDADES DE MEJORA

A continuación, se detallan oportunidades de mejora del código que surgen luego de una puesta a punto y análisis global del trabajo realizado.

1. La función mensajesAlerta(str1,str2) que se encuentra implicada en la confirmación o cancelación de acciones, no pudo ser tratada como una función pura. Luego de varios intentos fallidos, la función tuvo que escribirse 3 veces en total, una para cada caso en particular, por lo que hubo una redundancia de código. Dicha función puede encontrarse por ejemplo en modificarDisponibilidad().
2. Algunas de las funciones incluidas en Index.js, no aportan demasiado si se las compara con el código escrito en Funciones.js. Por ejemplo, la función estadisticaPersona() solo establece las ubicaciones en el HTML donde se imprimirán tablas estadísticas. La función invocada desde ésta es la que realmente realiza el trabajo de análisis e impresión, llevándose el mayor porcentaje de código. En una primera instancia, se estableció esta lógica para el trabajo que se mantuvo a lo largo del proyecto, de manera de establecer una consistencia en el código. Sin embargo, se llegó a la conclusión de que sería conveniente, en una nueva versión, eliminar este tipo de funciones
3. A lo largo del trabajo, una vez establecido y analizado todo el código, se concluyó que varios condicionales if podrían haber sido sustituidos por condicionales switch.