

Universidad ORT Uruguay

Facultad de Ingeniería

Escuela de Tecnología

OBLIGATORIO PROGRAMACIÓN 1



De Coster Florencia – 321005



Barbieri Paula - 327911

1 M C

Docente: Andrés de Sosa

Analista en Tecnologías de la Información

Fecha de entrega del documento: 19/10/2023

ÍNDICE

1. DOCUMENTO DE ANÁLISIS	3
1.1 Descripción general del problema a resolver.....	3
1.1.1 - Tipos de usuario del sistema	3
1.1.2 - Listado de funcionalidades.....	3
1.2 Detalle de Funcionalidades	4
1.2.1 F01 – Registro en la aplicación	4
1.2.2 F02 – Ingreso en la aplicación.....	6
1.2.3 F03 – Alquiler de una máquina virtual.....	7
1.2.4 F04 – Listado de máquinas alquiladas y su estado.....	9
1.2.5 F05 – Ver costo total del alquiler	10
1.2.6 F06 – Aprobación de registros de usuarios	11
1.2.7 F07 – Modificar stock de máquinas virtuales.....	12
1.2.8 F08 – Informe de máquinas virtuales	13
1.3 Tabla de casos de prueba	14
2 CÓDIGO COMPLETO	20
2.1 index.html	20
2.2 codigo.js	25
2.3 clases.js	34
2.4 estilos.css	45
3 LISTADO DE INFORMACIÓN PRECARGADA	52

1. DOCUMENTO DE ANÁLISIS

1.1 Descripción general del problema a resolver

Desarrollar un servicio de alquiler de máquinas virtuales en la nube que permita obtener temporalmente la capacidad de cómputo necesaria para correr modelos de Inteligencia Artificial. La aplicación deberá permitir la administración de dichas máquinas, así como también el alquiler de las mismas por parte de los usuarios.

1.1.1 - Tipos de usuario del sistema

Existen dos tipos de perfiles:

- **Usuario:** Este perfil está destinado a los usuarios de la aplicación, quienes podrán solicitar el alquiler de máquinas virtuales por determinado período de tiempo. Su función principal es utilizar el servicio y realizar solicitudes de alquiler.
- **Administrador:** Perfil encargado de la administración y gestión del servicio. Es importante destacar que los administradores no pueden ser creados a través de la aplicación, ya que deben ser preconfigurados en el sistema antes de su asignación. Sus funciones principales incluyen la supervisión, el mantenimiento y la toma de decisiones estratégicas relacionadas con el funcionamiento de la aplicación y los recursos virtuales disponibles.

1.1.2 - Listado de funcionalidades

F01 - Registro en la aplicación - Perfil: Usuario

F02 - Ingreso en la aplicación - Perfil: Usuario y Administrador

F03 - Alquiler de una máquina virtual - Perfil: Usuario

F04 - Ver listado de máquinas alquiladas y su estado - Perfil: Usuario

F05- Ver costo de alquiler - Perfil: Usuario

F06 - Aprobación de registro de usuarios - Perfil: Administrador

F07 - Modificar stock de máquinas virtuales - Perfil: Administrador

F08- Ver informe de máquinas virtuales - Perfil: Administrador

1.2 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.

1.2.1 F01 – Registro en la aplicación

1.2.1.1 ACCESO

Únicamente pueden registrarse los usuarios

1.2.1.2 DESCRIPCIÓN

El usuario se registra en la aplicación cargando datos: nombre, apellido, nombre de usuario, contraseña y número de tarjeta de crédito.

1.2.1.3 INTERFAZ DE USUARIO

The image shows a web form titled "REGISTRO" in green. It contains several input fields with labels above them: "NOMBRE:", "APELLIDO:", "NOMBRE DE USUARIO:", "CONTRASEÑA:", "TARJETA DE CRÉDITO:", and "CVC:". Below the password field, there is a text requirement: "Debe contener al menos 5 caracteres entre letras y números, una mayúscula y una minúscula." At the bottom of the form, there are two dark blue buttons with white text: "REGISTRARME" and "VOLVER".

1.2.1.4 VALIDACIONES

- En el ingreso de nombre y apellido, se debe validar que no sean o contengan números. En caso de incluir números, se mostrará un mensaje de error “ERROR: nombre y/o apellido incorrecto.”
- El nombre de usuario debe ser alfanumérico. En caso de no serlo se mostrará un mensaje “El nombre de usuario debe contener al menos una letra y un número.”
- El nombre de usuario debe ser único, por lo que si se elige un usuario ya creado se muestra en pantalla “Usuario ya utilizado, intente con otra opción.”
- La contraseña debe tener al menos 5 caracteres, contando con al menos 1 mayúscula, 1 minúscula y un número. En caso de no cumplir las validaciones se muestra “Contraseña inválida”.
- El número de tarjeta de crédito debe tener un formato de “xxxx-xxxx-xxxx-xxxx”. Los dígitos ingresados deben tener un largo de 16 caracteres sin contar guiones y que el número de tarjeta sea un número válido corroborando su dígito verificador mediante el algoritmo de verificación de tarjetas. De no ser correcto se mostrará un mensaje “Tarjeta de crédito incorrecta. Ingrese nuevamente”.
- El CVC debe contener tres números. En caso de contrario se muestra el mensaje: “El código CVC debe contener 3 números, los encuentra en el dorso de la tarjeta”.

Todos los datos solicitados son obligatorios para completar el registro, en caso de dejar alguno vacío, se muestra un mensaje “Todos los datos deben ser completados”.

1.2.2 F02 – Ingreso en la aplicación

1.2.2.1 ACCESO

Pueden acceder tanto usuarios como administradores

1.2.2.2 DESCRIPCIÓN

El usuario ingresará en la aplicación siempre y cuando, previamente haya completado su registro y el administrador lo haya aprobado. Se le solicitará nombre de usuario y contraseña.

En el caso de los administradores, deberán ingresar usuario y contraseña que fueron previamente cargados en el sistema.

1.2.2.3 INTERFAZ DE USUARIO



The screenshot displays a login interface with a light gray background. At the top, the word "BIENVENIDO" is written in green. Below it, the title "Inicio de Sesión" is in dark blue. There are two input fields: the first is labeled "NOMBRE DE USUARIO:" and the second is labeled "CONTRASEÑA:". Below the password field is a dark blue button labeled "INGRESAR". At the bottom, there is a link "¿USUARIO NUEVO? PODÉS REGISTRARTE HACIENDO CLICK" followed by a dark blue button labeled "REGISTRARME COMO NUEVO USUARIO".

1.2.2.4 VALIDACIONES

- Si el nombre de usuario y/o contraseña son vacíos o incorrectos (no cumple con las condiciones anteriormente mencionadas o no coinciden con los datos anteriormente registrados), se muestra “Usuario o contraseña incorrectos. Intente nuevamente”.
- Si los datos coinciden con los almacenados en el sistema, y el usuario o administrador se encuentra habilitado, se despliega el menú correspondiente al perfil.
- Si el usuario se encuentra bloqueado o aún en estado pendiente de aprobación, no podrá acceder y se muestra el mensaje: “Usuario no habilitado”.

1.2.3 F03 – Alquiler de una máquina virtual

1.2.3.1 ACCESO

Pueden acceder solamente usuarios.

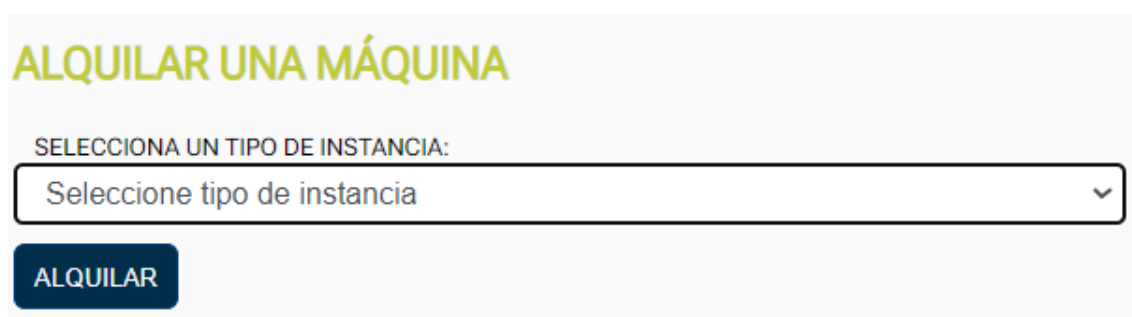
1.2.3.2 DESCRIPCIÓN

Los usuarios tienen la opción de alquilar máquinas virtuales para su utilización. Este servicio conlleva un costo de alquiler asociado.

Cuando un usuario alquila una máquina virtual, esta se enciende automáticamente y, en la primera ocasión, no se le cobra por la activación; únicamente se factura el alquiler.

Para llevar a cabo el proceso de alquiler, el usuario deberá elegir el tipo de instancia que desea alquilar de un combo desplegable con la información mostrada en la interfaz de usuario.

1.2.3.3 INTERFAZ DE USUARIO

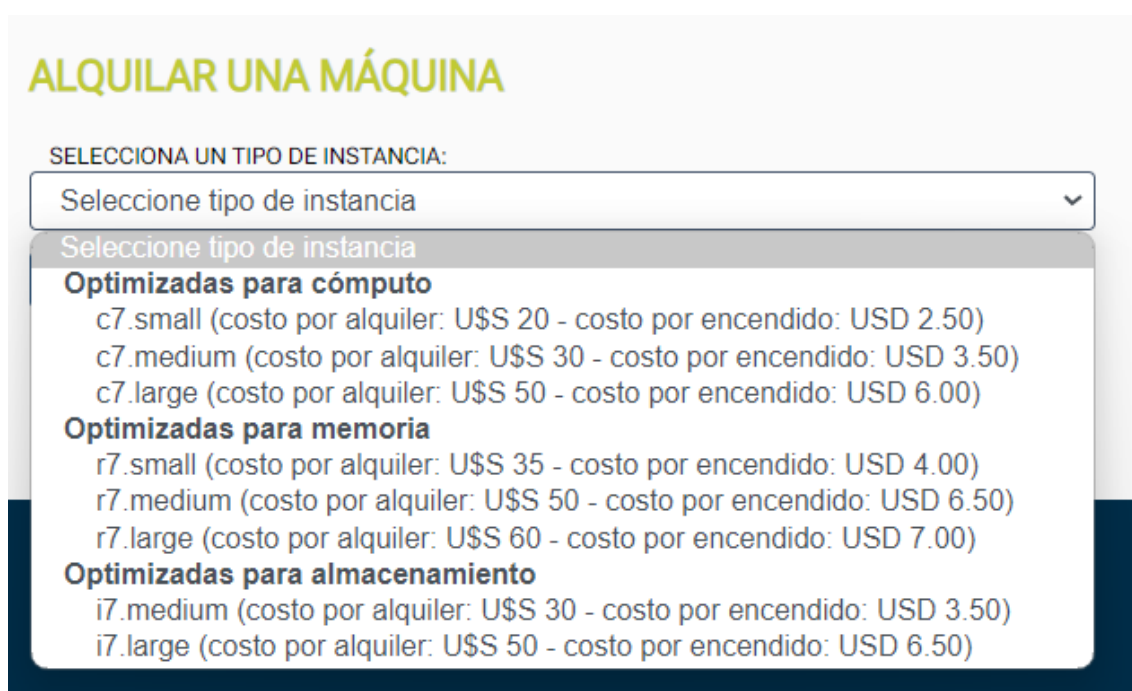


ALQUILAR UNA MÁQUINA

SELECCIONA UN TIPO DE INSTANCIA:

Seleccione tipo de instancia ▼

ALQUILAR



ALQUILAR UNA MÁQUINA

SELECCIONA UN TIPO DE INSTANCIA:

Seleccione tipo de instancia ▼

Seleccione tipo de instancia

Optimizadas para cómputo

- c7.small (costo por alquiler: U\$S 20 - costo por encendido: USD 2.50)
- c7.medium (costo por alquiler: U\$S 30 - costo por encendido: USD 3.50)
- c7.large (costo por alquiler: U\$S 50 - costo por encendido: USD 6.00)

Optimizadas para memoria

- r7.small (costo por alquiler: U\$S 35 - costo por encendido: USD 4.00)
- r7.medium (costo por alquiler: U\$S 50 - costo por encendido: USD 6.50)
- r7.large (costo por alquiler: U\$S 60 - costo por encendido: USD 7.00)

Optimizadas para almacenamiento

- i7.medium (costo por alquiler: U\$S 30 - costo por encendido: USD 3.50)
- i7.large (costo por alquiler: U\$S 50 - costo por encendido: USD 6.50)

1.2.3.4 **VALIDACIONES**

- Debe seleccionar obligatoriamente una opción, de lo contrario no se podrá continuar con el proceso de alquiler. Se muestra el mensaje: "No fue posible realizar el alquiler."
- Para la opción seleccionada, se debe chequear la disponibilidad de stock. En caso de no contar con la máquina solicitada, se mostrará el mismo mensaje "No fue posible realizar el alquiler."

1.2.4 F04 – Listado de máquinas alquiladas y su estado

1.2.4.1 ACCESO

Pueden acceder solamente usuarios.

1.2.4.2 DESCRIPCIÓN

Los usuarios pueden acceder a una lista de sus máquinas alquiladas, donde pueden ver el tipo de instancia, su estado, las veces que se inició la instancia, la opción de encender o apagar la instancia (con su respectivo costo), y un filtro con las instancias según su estado.

1.2.4.3 INTERFAZ DE USUARIO

MIS MAQUINAS ALQUILADAS

Filtrar instancias ▼

Filtrar instancias

- Todas las instancias
- Instancias Apagadas
- Instancias Encendidas

MIS MAQUINAS ALQUILADAS

Todas las instancias ▼

Instancia	Estado	Veces Iniciada	Acción
c7.large	encendido	0	CAMBIAR
c7.large	encendido	0	CAMBIAR
c7.medium	encendido	0	CAMBIAR

VALIDACIONES

- En caso de no tener máquinas alquiladas, se oculta la tabla y se muestra el mensaje “No hay alquileres.”
- Para cada opción seleccionada en el combo desplegable tienen que mostrarse en pantalla, dentro de la tabla, únicamente los alquileres de máquinas en el estado solicitado. De no haber máquinas en ese estado, se muestra el mensaje “No hay alquileres”.

1.2.5 F05 – Ver costo total del alquiler

1.2.5.1 ACCESO

Pueden acceder solamente usuarios.

1.2.5.2 DESCRIPCIÓN

El usuario deberá poder visualizar el costo total por el uso de sus máquinas alquiladas en el momento. En esta pantalla muestra en detalle los gastos totales agrupados por tipo de instancia.

1.2.5.3 INTERFAZ DE USUARIO

MIS GASTOS			
Tipo de instancia	Costo por encendido	Veces encendidas	Gasto total
c7.medium	U\$S 3.50	0	U\$S 30
c7.large	U\$S 6	0	U\$S 100

1.2.5.4 VALIDACIONES

- Para este cálculo no se debe contar el encendido inicial de cada instancia que es “gratuito”.
- Además, se deben agrupar en una única fila todos los alquileres de un mismo tipo de instancia.

1.2.6 F06 – Aprobación de registros de usuarios

1.2.6.1 ACCESO

Pueden acceder solamente administradores.

1.2.6.2 DESCRIPCIÓN

Los administradores tendrán la capacidad de visualizar la lista de usuarios que se han registrado en el sistema, incluyendo aquellos cuyos registros están pendientes de aprobación, así como aquellos que ya han sido aprobados. Además, podrán modificar el estado de los usuarios en cualquier momento.

Si un usuario se encuentra en estado "pendiente", los administradores pueden cambiar su estado a "activo", permitiendo así acceder al sistema y utilizar sus cuentas.

Los administradores también tienen la capacidad de bloquear a los usuarios activos. Al realizar esta acción y cambiar el estado de un usuario a "bloqueado".

1.2.6.3 INTERFAZ DE USUARIO

ESTADO DE USUARIOS		
Nombre de usuario	Estado	Acción
usu1	activo	CAMBIAR
usu2	activo	CAMBIAR
usu3	activo	CAMBIAR
usu4	bloqueado	CAMBIAR
usu5	pendiente	CAMBIAR

1.2.6.4 VALIDACIONES

Los usuarios bloqueados quedarán inhabilitados para iniciar sesión y todas las máquinas alquiladas por los mismos volverán a estar disponibles para su alquiler, eliminando sus alquileres.

1.2.7 F07 – Modificar stock de máquinas virtuales

1.2.7.1 ACCESO

Pueden acceder solamente administradores

1.2.7.2 DESCRIPCIÓN

Los administradores podrán modificar el stock de máquinas virtuales disponibles por tipo de instancia.

1.2.7.3 INTERFAZ DE USUARIO



The image displays two versions of the 'GESTIÓN DE STOCK' (Stock Management) user interface. The left version shows the form with 'c7.medium' selected in the instance type dropdown, the quantity '9' entered in the stock input field, and '+' and '-' buttons for adjusting the stock. The right version shows the same form with the dropdown menu open, revealing a list of available instance types: 'c7.small', 'c7.medium' (which is highlighted in blue), 'c7.large', 'r7.small', 'r7.medium', 'r7.large', 'i7.medium', and 'i7.large'.

1.2.7.4 VALIDACIONES

- El stock solamente puede ser disminuido hasta la cantidad de máquinas alquiladas actualmente. En este último caso, haciendo click en el botón “-”, “No es posible realizar la modificación de Stock, no puede tener menos stock que máquinas en alquiler.”

1.2.8 F08 – Ver informe de máquinas virtuales

1.2.8.1 ACCESO

Pueden acceder solamente administradores.

1.2.8.2 DESCRIPCIÓN

Los administradores tendrán la posibilidad de acceder a una pantalla de informes especialmente diseñada, donde podrán contemplar una tabla informativa acerca de tipos de instancia en alquiler, cantidad y el ingreso total generado hasta la fecha por cada tipo. Además, en esta misma pantalla, se dispondrá un párrafo que resaltará el ingreso total.

1.2.8.3 INTERFAZ DE USUARIO

INFORME DE MÁQUINAS VIRTUALES

Tipo de instancia	Cantidad en Alquiler	Ingreso Total Actual
c7.medium	2	U\$S 60
c7.large	2	U\$S 100
r7.small	1	U\$S 35
r7.medium	2	U\$S 100
i7.medium	1	U\$S 30
i7.large	2	U\$S 100

El total de ingresos es: U\$S 425

1.2.8.4 VALIDACIONES

- No hay validaciones ya que no se le solicita nada al administrador, simplemente se le otorga información.

1.3 Tabla de casos de prueba

N° prueba	Funcionalidad	Detalle	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
1	Registro de usuario	Usuario ingresa los datos: Nombre, Apellido, Nombre de usuario, Contraseña y número de tarjeta de crédito	Ingresa algún dato vacío	Se muestra el mensaje: “Todos los datos deben ser completados” y no se puede registrar.	Se muestra el mensaje: “Todos los datos deben ser completados” y no se puede registrar.	P
2	Registro de usuario	“	Ingreso de nombre o apellido conteniendo al menos un número	Se muestra mensaje: “ERROR: Nombre o apellido incorrecto” y no se puede registrar.	Se muestra mensaje: “ERROR: Nombre o apellido incorrecto” y no se puede registrar.	P
3	Registro de usuario	“	Ingreso de nombre de usuario sin números	Se muestra el mensaje: “El nombre de usuario debe contener al menos una letra y un número” y no se puede registrar.	Se muestra el mensaje: “El nombre de usuario debe contener al menos una letra y un número” y no se puede registrar.	P
4	Registro de usuario	“	Ingreso un nombre de usuario que ya exista	Se muestra mensaje: “Nombre de usuario ya utilizado” y no se puede registrar.	Se muestra mensaje: “Nombre de usuario ya utilizado” y no se puede registrar.	P

5	Registro de usuario	“	Contraseña con menos de 5 caracteres	Se muestra mensaje: “Contraseña inválida” y no se puede registrar.	Se muestra mensaje: “Contraseña inválida” y no se puede registrar.	P
6	Registro de usuario	“	Contraseña sin mayúsculas	Se muestra mensaje: “Contraseña inválida” y no se puede registrar.	Se muestra mensaje: “Contraseña inválida” y no se puede registrar.	P
7	Registro de usuario	“	Ingresar número de tarjeta sin formato correcto	Se muestra el mensaje: “Tarjeta de crédito incorrecta. Ingrese nuevamente” y no se puede registrar.	Se muestra el mensaje: “Tarjeta de crédito incorrecta. Ingrese nuevamente” y no se puede registrar.	P
8	Ingreso en la aplicación	Se solicita nombre de usuario y contraseña	Ingreso de usuario o contraseña que no coinciden con los registrados	Se muestra mensaje: “Usuario o contraseña incorrectos. Intente nuevamente.” y no se puede acceder.	Se muestra mensaje: “Usuario o contraseña incorrectos. Intente nuevamente.” y no se puede acceder.	P
9	Ingreso en la aplicación	“	Ingreso de usuario o contraseña vacíos	Se muestra mensaje: “Usuario o contraseña incorrectos. Intente nuevamente.” y no se puede acceder.	Se muestra mensaje: “Usuario o contraseña incorrectos. Intente nuevamente.” y no se puede acceder.	P
10	Ingreso en la aplicación	“	Ingreso de usuario pendiente de aprobación	Mensaje en pantalla: “Usuario no habilitado” y no puede acceder.	Mensaje en pantalla: “Usuario no habilitado” y no puede acceder.	P

11	Ingreso en la aplicación	“	Ingreso de usuario bloqueado	Mensaje en pantalla: “Usuario no habilitado” y no puede acceder.	Mensaje en pantalla: “Usuario no habilitado” y no puede acceder.	P
12	Ingreso en la aplicación	“	Ingreso de datos correctos, usuario habilitado	Se despliega el menú correspondiente al perfil.	Se despliega el menú correspondiente al perfil.	P
13	Alquiler de una máquina	Se despliega un combo de opciones para que el usuario clickee una	Selección de instancia sin stock	Se muestra el mensaje “No fue posible realizar el alquiler” y no se alquila.	Se muestra el mensaje “No fue posible realizar el alquiler” y no se alquila.	P
14	Alquiler de una máquina	“	Seleccionar una máquina en stock	Se muestra el mensaje “Alquiler exitoso!” Agregar el alquiler, a la lista de objetos alquileres. Tiene que sumarse uno a la propiedad cantidad alquiladas del objeto instancia.	Se muestra el mensaje “Alquiler exitoso!” Se agrega el alquiler, a la lista de objetos alquileres. Se suma uno a la propiedad cantidad alquiladas del objeto instancia.	P
15	Alquiler de una máquina	El usuario no selecciona una opción del combo desplegable	Hace click en el botón “Alquilar”.	Se muestra el mensaje “No fue posible realizar el alquiler” y no se alquila.	Se muestra el mensaje “No fue posible realizar el alquiler” y no se alquila.	

16	Ver listado de máquinas alquiladas	Seleccionar del combo “Todas las instancias”	No tiene máquinas alquiladas aún	Se oculta la tabla y se muestra el mensaje “No hay alquileres.”	Se ocultó la tabla y se mostró el mensaje “No hay alquileres.”	P
17	Ver listado de máquinas alquiladas	Seleccionar del combo “Todas las instancias”	Tiene máquinas alquiladas encendidas y apagadas	Se muestra tabla con instancias encendidas y apagadas.	Se mostró la tabla con instancias encendidas y apagadas.	P
18	Ver listado de máquinas alquiladas	Seleccionar del combo “Apagadas”	Tengo máquinas alquiladas encendidas, pero no apagadas	Se oculta la tabla y se muestra el mensaje “No hay alquileres.”	Se ocultó la tabla y se mostró el mensaje “No hay alquileres.”	P
19	Ver listado de máquinas alquiladas	Seleccionar del combo “Encendidas”	Tengo máquinas alquiladas apagadas, pero ninguna encendida	Se oculta la tabla y se muestra el mensaje “No hay alquileres.”	Se ocultó la tabla y se mostró el mensaje “No hay alquileres.”	P
20	Ver listado de máquinas alquiladas	Seleccionar del combo “Apagadas”	Tengo máquinas apagadas y encendidas	Se muestra tabla SOLAMENTE con las apagadas	Se muestra tabla SOLAMENTE con las apagadas	P
21	Ver listado de máquinas alquiladas	Seleccionar del combo “Encendidas”	Tengo máquinas apagadas y encendidas	Se muestra tabla SOLAMENTE con las encendidas	Se muestra tabla SOLAMENTE con las encendidas	P
22	Aprobación de registro de usuarios	Usuario pendiente	Click en botón “Cambiar”	Se activa el usuario y su estado pasa a “Activo” El usuario puede loguearse.	Se activa el usuario y su estado pasa a “Activo” El usuario pudo loguearse.	P

23	Aprobación de registro de usuarios	Usuario pendiente a bloqueado	Click en botón "Cambiar"	Se debe cambiar a activo para luego poder cambiar a bloqueado.	Se debe cambiar a activo para luego poder cambiar a bloqueado.	P
24	Aprobación de registro de usuarios	Usuario activo a bloqueado	Click en botón "Cambiar"	Debe bloquear al usuario para que éste no pueda loguearse. Se eliminan sus alquileres, volviendo a poner esas máquinas disponibles para alquilar	Usuario bloqueado, no se pudo loguear Se eliminaron sus alquileres, volviendo a poner esas máquinas disponibles para alquilar	P
25	Aprobación de registro de usuarios	Usuario bloqueado a activo	Click en botón "Cambiar"	Debe activar al usuario. Este podrá iniciar sesión y volver a alquilar máquinas desde cero.	Se activó al usuario. Pudo iniciar sesión y volver a alquilar máquinas desde cero.	P
26	Modificación de stock de máquinas virtuales	Disminuir stock	Disminuir el stock hasta un valor menor que la cantidad de máquinas alquiladas haciendo click en "-".	Cuando la cantidad de máquinas sea la misma que las alquiladas, se muestra el mensaje "No es posible realizar la modificación de Stock, no puede tener menos stock que máquinas en alquiler." y no se puede disminuir más.	Se mostró el mensaje "No es posible realizar la modificación de Stock, no puede tener menos stock que máquinas en alquiler." y no se puede disminuir más.	P
27	Modificación de stock de máquinas virtuales	Aumentar stock	Se puede aumentar la cantidad que se desee con el botón "+"	Botón "+" siempre habilitado. Se aumentará el stock.	Botón "+" siempre habilitado. Se aumentó el stock.	P

28	Informe de máquinas	Visualizar tabla de informe con los alquileres. Alquileres agrupados por tipo de instancia.	Existen alquileres generados, pero NO de todos los tipos de instancias	No se deben mostrar en la tabla los tipos de instancias sin alquiler.	No se mostraron los tipos instancia sin alquileres.	P
29	Informe de máquinas	Visualizar tabla de informe con los alquileres.	No hay alquileres de ningún tipo de instancia	No se debe mostrar tabla	No se muestra la tabla.	P

2 CÓDIGO COMPLETO

2.1 index.html

```
1. <!DOCTYPE html>
2. <html lang="es">
3.   <head>
4.     <meta charset="UTF-8">
5.     <meta name="viewport" content="width=device-width, initial-
      scale=1.0">
6.     <script src="js/clases.js"></script>
7.     <script src="js/codigo.js"></script>
8.     <link rel="stylesheet" href="css/estilos.css" />
9.     <title>Alquiler de Máquinas Virtuales</title>
10.  </head>
11.  <body>
12.    <div id="contenedor">
13.      <div id="cabezal" class="clearfix">
14.        
15.        <!-- Menú usuario -->
16.        <ul id="menuUsuario">
17.          <li>
18.            <a id="menuAlquilarMaquina">Alquilar una Máquina</a>
19.          </li>
20.          <li>
21.            <a id="menuVerAlquiler">Mis alquileres</a>
22.          </li>
23.          <li>
24.            <a id="menuMisGastos">Mis gastos</a>
25.          </li>
26.          <li>
27.            <a id="menuSalir">Salir</a>
28.          </li>
29.        </ul>
30.        <!-- Menú administrador -->
31.        <ul id="menuAdmin">
32.          <li>
33.            <a id="listarUsuarios">Listar Usuarios</a>
34.          </li>
35.          <li>
36.            <a id="menuStock">Stock</a>
37.          </li>
38.          <li>
39.            <a id="informeMaquinas">Informes de máquinas</a>
40.          </li>
41.          <li>
42.            <a id="menuSalirAdmin">Salir</a>
43.          </li>
44.        </ul>
```

```

45.     </div>
46.     <!-- FIN CABEZAL -->
47.     <div id="contenido">
48.         <div id="InicioSesión">
49.             <h2>Bienvenido</h2>
50.             <h3>Inicio de Sesión</h3>
51.             <!-- INPUTS -->
52.             <label for="nombreUsuario">Nombre de usuario: </label>
53.             <input id="nombreUsuario" type="text"><br>
54.             <label for="clave">Contraseña: </label>
55.             <input id="clave" type="password"><br>
56.             <input type="button" id="ingresar" value="Ingresar"><br><br>
57.             <p id="mensajesLogin"></p><br><br>
58.
59.             <label for="btnIrAregistro">¿Usuario nuevo? Podés
registrarte haciendo click </label>
60.             <input type="button" id="btnIrAregistro" value="Regístrame
como nuevo usuario">
61.         </div>
62.         <!-- Registro usuario -->
63.         <div id="registroUsuario">
64.             <h2>Registro</h2>
65.             <label for="nombre">Nombre: </label> <input id="nombre"
type="text"><br>
66.             <label for="apellido">Apellido: </label> <input id="apellido"
type="text"><br>
67.             <label for="registroNombreUsuario">Nombre de usuario:
</label> <input id="registroNombreUsuario" type="text"><br>
68.             <label for="contraseña">Contraseña: </label>
69.             <input id="contraseña" type="password">
70.             <p>Debe contener al menos 5 caracteres entre letras y números,
una mayúscula y una minúscula.</p><br>
71.             <label for="tarjetaCredito">Tarjeta de crédito:</label>
72.             <input id="tarjetaCredito" type="text">
73.             <label for="CVCTarjetaCredito">CVC:</label>
74.             <input id="CVCTarjetaCredito" type="number"><br><br>
75.             <input type="button" id="regístrame"
value="Regístrame"><br><br>
76.             <p id="msjRegistroUsuario"></p> <br>
77.             <input type="button" id="volver" value="Volver">
78.         </div>
79.
80.         <div id="registroExitoso">
81.             <p id="finregistro">Gracias por tu registro, debes esperar a
que un administrador te habilite</p>
82.             <input type="button" id="volverAlInicio"
value="Salir"><br><br>
83.         </div>
84.

```

```

85.         <!-- Alquilar máquinas -->
86.         <div id="selectInstancias">
87.             <h2>Alquilar una máquina</h2>
88.             <label for="tipoInstancia">Selecciona un tipo de
instancia:</label>
89.             <select id="tipoInstancia" name="tipoInstancia">
90.                 <option value="" disabled selected>Seleccione tipo de
instancia</option>
91.                 <optgroup label="Optimizadas para cómputo">
92.                     <option value="c7s">c7.small (costo por alquiler: U$S
20 - costo por encendido: USD 2.50)</option>
93.                     <option value="c7m">c7.medium (costo por alquiler: U$S
30 - costo por encendido: USD 3.50)</option>
94.                     <option value="c7l">c7.large (costo por alquiler: U$S
50 - costo por encendido: USD 6.00)</option>
95.                 </optgroup>
96.                 <optgroup label="Optimizadas para memoria">
97.                     <option value="r7s">r7.small (costo por alquiler: U$S
35 - costo por encendido: USD 4.00)</option>
98.                     <option value="r7m">r7.medium (costo por alquiler: U$S
50 - costo por encendido: USD 6.50)</option>
99.                     <option value="r7l">r7.large (costo por alquiler: U$S
60 - costo por encendido: USD 7.00)</option>
100.                </optgroup>
101.                <optgroup label="Optimizadas para almacenamiento">
102.                    <option value="i7m">i7.medium (costo por
alquiler: U$S 30 - costo por encendido: USD 3.50)</option>
103.                    <option value="i7l">i7.large (costo por
alquiler: U$S 50 - costo por encendido: USD 6.50)</option>
104.                </optgroup>
105.            </select>
106.            <input type="button" id="alquilar"
value="Alquilar"><br><br>
107.            <p id="mensajeAlquiler"></p>
108.        </div>
109.
110.        <!-- Filtrar máquinas alquiladas -->
111.        <div id="maquinasAlquiladas">
112.            <h2>Mis máquinas alquiladas</h2>
113.            <select id="filtroApagadoEncendido">
114.                <option value="" selected>Seleccionar filtro</option>
115.                <option value="T">Todas las instancias</option>
116.                <option value="A">Instancias Apagadas</option>
117.                <option value="E">Instancias Encendidas</option>
118.            </select>
119.
120.            <!-- TABLAS -->
121.            <table id="tablaInstanciasFiltro"></table>
122.            <p id="textoInstancias"></p>

```

```

123.         </div>
124.
125.         <div id="costosAlquiler">
126.             <h2>Mis gastos</h2>
127.             <table id="tablaCostos"></table>
128.             <p id="textoCostos"></p>
129.         </div>
130.
131.         <div id="tablaDeUsuarios">
132.             <h2>Estado de usuarios</h2>
133.             <table id="tablaUsuarios"></table>
134.             <p id="textoTablaUsuarios"></p>
135.         </div>
136.
137.         <!-- STOCK -->
138.         <div id="stock">
139.             <h2>Gestión de stock</h2>
140.             <select id="selectStock">
141.                 <option value="" disabled selected>Seleccione tipo de
instancia</option>
142.                 <option value="c7s">c7.small</option>
143.                 <option value="c7m">c7.medium</option>
144.                 <option value="c7l">c7.large</option>
145.                 <option value="r7s">r7.small</option>
146.                 <option value="r7m">r7.medium</option>
147.                 <option value="r7l">r7.large</option>
148.                 <option value="i7m">i7.medium</option>
149.                 <option value="i7l">i7.large</option>
150.             </select>
151.             <input type="number" id="cantStock" value="">
152.             <input type="button" id="incrementarStock" value="+">
153.             <input type="button" id="disminuirStock" value="-
"><br><br>
154.             <p id="msjStock"></p>
155.
156.         </div>
157.
158.         <div id="informeDeMaquinas">
159.             <h2>Informe de máquinas virtuales</h2>
160.             <table id="tablaInformeMaquinas"></table>
161.             <p id="textoInformeMaquinas"></p>
162.         </div>
163.     </div>
164.     <!-- FIN CONTENIDO -->
165.
166.     <div id="footer">
167.         
168.         <h4 id="footerNombres">Paula Barbieri y Florencia De
Coster</h4>

```

```
169.         <h4 id="footerDos">PROGRAMACIÓN 1 | ANALISTA EN TI |  
2023</h5>  
170.         <h4 id="footerTres">ORT</h5>  
171.     </div>  
172.     <div style="clear:both;"></div>  
173.     <!-- FIN PIE -->  
174. </div>  
175.     <!-- FIN CONTENEDOR -->  
176. </body>  
177. </html>  
178.
```


2.2 codigo.js

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

function inicio(){
    ocultarFormularios();
    document.querySelector("#InicioSesión").style.display= "block"
    document.querySelector("#btnIrAregistro").addEventListener("click",
registroUsuario);
    document.querySelector("#registrarme").addEventListener("click",
verificarRegistro);
    document.querySelector("#volverAlInicio").addEventListener("click",
volverAlInicio);
    document.querySelector("#volver").addEventListener("click",
volverAlInicio);
    document.querySelector("#menuSalirAdmin").addEventListener("click", volverA
lInicio);
    document.querySelector("#menuSalir").addEventListener("click",
volverAlInicio);
    document.querySelector("#ingresar").addEventListener("click", ingresar);
    document.querySelector("#listarUsuarios").addEventListener("click",
listarUsuarios);
    document.querySelector("#menuAlquilarMaquina").addEventListener("click",
mostrarAlquilerMaquina);
    document.querySelector("#alquilar").addEventListener("click",
alquilerMaquina); // boton para hacer efectivo el alquiler
    document.querySelector("#tipoInstancia").addEventListener("click",
elegirMaquinaParaStock); // click del select
    document.querySelector("#menuMisGastos").addEventListener("click",
irATablaGastos);
    document.querySelector("#menuStock").addEventListener("click",
mostrarStock);
    document.querySelector("#selectStock").addEventListener("change",
modificarStockSelect);
    document.querySelector("#incrementarStock").addEventListener("click",
sumarStock);
    document.querySelector("#disminuirStock").addEventListener("click",
restarStock);
    document.querySelector("#menuVerAlquiler").addEventListener("click",
irATablaAlquiler);
    document.querySelector("#filtroApagadoEncendido").addEventListener("change
", tablaAlquiler); //click del select
    document.querySelector("#informeMaquinas").addEventListener("click",
irATablaGanancias);

    miSistema.precargarDatos();
}

//Ocultar formularios
```

```

function ocultarFormularios(){
    document.querySelector("#menuUsuario").style.display= "none";
    document.querySelector("#menuAdmin").style.display= "none";
    document.querySelector("#registroUsuario").style.display= "none";
    document.querySelector("#registroExitoso").style.display= "none";
    document.querySelector("#InicioSesión").style.display= "none";
    document.querySelector("#selectInstancias").style.display= "none";
    document.querySelector("#maquinasAlquiladas").style.display= "none";
    document.querySelector("#costosAlquiler").style.display= "none";
    document.querySelector("#tablaDeUsuarios").style.display= "none";
    document.querySelector("#maquinasAlquiladas").style.display= "none";
    document.querySelector("#stock").style.display= "none";
    document.querySelector("#informeDeMaquinas").style.display= "none";
}

//REGISTRO DE USUARIO
function registroUsuario(){
    ocultarFormularios();
    document.querySelector("#registroUsuario").style.display= "block";
}

function verificarRegistro(){
    let nombre= document.querySelector("#nombre").value;
    let apellido = document.querySelector("#apellido").value;
    let nombreUsuario =
document.querySelector("#registroNombreUsuario").value;
    let contraseña = document.querySelector("#contraseña").value;
    let tarjeta = document.querySelector("#tarjetaCredito").value;
    let nombreApellidoValido= miSistema.verificarNombreyApellido(nombre,
apellido);
    let nombreUsuarioValido = miSistema.verificarNombreUsuario(nombreUsuario);
    let nombreUsuarioRepetido =
miSistema.verificarNombreUsuarioRepetido(nombreUsuario);
    let contraseñaValida = miSistema.verificarContraseñaValida(contraseña);
    let tarjetaValida = miSistema.algoritmoLuhn(tarjeta);
    let CVCTarjetaCredito =document.querySelector("#CVCTarjetaCredito").value;

    document.querySelector("#msjRegistroUsuario").innerHTML="";

    if(nombre==" || apellido==" || nombreUsuario==" || contraseña==" ||
tarjeta==" || CVCTarjetaCredito==""){
        document.querySelector("#msjRegistroUsuario").innerHTML = "Todos los
datos deben ser completados."
    } else {
        if (nombreApellidoValido == false) {
            document.querySelector("#msjRegistroUsuario").innerHTML += "<br><br>
ERROR: nombre y/o apellido incorrecto<br><br>"
        }
        if (nombreUsuarioValido == false) {

```

```

        document.querySelector("#msjRegistroUsuario").innerHTML +=
"<br><br> El nombre de usuario debe contener al menos una letra y un número."
    }
    if (nombreUsuarioRepetido == true) {
        document.querySelector("#msjRegistroUsuario").innerHTML +=
"<br><br> Usuario ya utilizado, intente con otra opción."
    }
    if (contraseñaValida == false) {
        document.querySelector("#msjRegistroUsuario").innerHTML +=
"<br><br> Contraseña inválida."
    }
    if (tarjetaValida == false) {
        document.querySelector("#msjRegistroUsuario").innerHTML
+= "<br> Tarjeta de crédito incorrecta. Ingrese nuevamente"
    }
    if (isNaN(CVCtarjetaCredito) || (CVCtarjetaCredito.length) != 3) {
        document.querySelector("#msjRegistroUsuario").innerHTML += "<br> El
código CVC debe contener 3 números, los encuentra en el dorso de la tarjeta."
    }
}
if (nombreApellidoValido && nombreUsuarioValido && nombreUsuarioRepetido
== false && contraseñaValida && tarjetaValida && !isNaN(CVCtarjetaCredito) &&
CVCtarjetaCredito.length == 3) {
    miSistema.agregarUsuario(nombreUsuario, contraseña, nombre, apellido,
tarjeta);
    ocultarFormularios();
    document.querySelector("#registroExitoso").style.display = "block";
}
}
function ingresar(){
    let usuario = document.querySelector("#nombreUsuario").value;
    let contraseña = document.querySelector("#clave").value;
    let tipoUsuario = miSistema.revisarUsuarioContraseña(usuario, contraseña);
    if (tipoUsuario == "usuario") { // si se pudo logear y es usuario muestro
menú del usuario
        ocultarFormularios();
        document.querySelector("#menuUsuario").style.display = "block";
    }
    if (tipoUsuario == "admin") { // si se pudo logear y es administrador se
muestra menú del administrador
        ocultarFormularios();
        document.querySelector("#menuAdmin").style.display = "block";
    }
    if (tipoUsuario == "incorrecto") { // no pudo logearse
        document.querySelector("#mensajesLogin").innerHTML = "Usuario o
contraseña incorrectos. Intente nuevamente.";
    }
}

```

```

        if(tipoUsuario == "Usuario no habilitado"){//está pendiente o bloqueado
            document.querySelector("#mensajesLogin").innerHTML = "Usuario no
            habilitado.";
        }
    }

function volverAlInicio(){
    ocultarFormularios();
    document.querySelector("#nombreUsuario").value="";
    document.querySelector("#clave").value="";
    document.querySelector("#mensajesLogin").innerHTML = "";
    document.querySelector("#InicioSesión").style.display= "block";
}

//GESTIONAR ESTADO DE USUARIOS
function listarUsuarios(){
    ocultarFormularios();
    document.querySelector("#menuAdmin").style.display= "block";
    document.querySelector("#tablaDeUsuarios").style.display= "block";

    let lista = miSistema.obtenerUsuarioComun();// la lista tiene objetos
    usuarios solamente
    let mensaje = "";
    if (lista.length>0){ // Tiene elementos
        let miTabla = document.querySelector("#tablaUsuarios");
        miTabla.innerHTML = "";
        tablaHTML = `<tr><th>Nombre de
usuario</th><th>Estado</th><th>Acción</th></tr>`
        mensaje="";

        for (let pos=0;pos<lista.length;pos++){
            let objUsuario = lista[pos];
            tablaHTML+=`<tr><td>${objUsuario.nombreUsuario}</td>
<td>${objUsuario.estado}</td>
<td><input class="button" type="button"
id="${objUsuario.nombreUsuario}" value="Cambiar"></td>
</tr>`
        }
        miTabla.innerHTML+= tablaHTML;

        for (let pos=0;pos<lista.length;pos++){
            let objUsuario = lista[pos];
            document.querySelector("#"+
objUsuario.nombreUsuario).addEventListener("click", cambiarEstado)
        }
    }
}

```

```

        else {
            mensaje = "No hay datos para procesar (no hay usuarios de tipo
Público).";
        }
        document.querySelector("#textoTablaUsuarios").innerHTML = mensaje;
    }
function cambiarEstado(){
    let idUsuario= this.id
    miSistema.cambiarEstado(idUsuario);
    listarUsuarios();
}

// GESTION DE STOCK

let cant = 0; // Variable global para mantener el contador de stock

function mostrarStock() {
    ocultarFormularios();
    document.querySelector("#menuAdmin").style.display = "block";
    document.querySelector("#stock").style.display = "block";
}
function elegirMaquinaParaStock(){
    modificarStock("selectStock");
    document.querySelector("#cantStock").value = "";
    document.querySelector("#cantStock").innerHTML= cant;
}
function modificarStockSelect(){
    modificarStock("selectStock")
    document.querySelector("#cantStock").value = cant;
}
function modificarStock(idSelect){
    let maquinaElegida = document.querySelector("#"+idSelect).value;
    let pos = 0;
    // Actualizar el valor de 'cant' según la máquina elegida
    switch (maquinaElegida) {
        case "c7s":
            cant = miSistema.listaInstancias[0].stock;
            pos=0;
            break;
        case "c7m":
            cant = miSistema.listaInstancias[1].stock;
            pos=1
            break;
        case "c7l":
            cant = miSistema.listaInstancias[2].stock;
            pos=2;
            break;
        case "r7s":
            cant = miSistema.listaInstancias[3].stock;

```

```

        pos=3;
        break;
    case "r7m":
        cant = miSistema.listaInstancias[4].stock;
        pos=4;
        break;
    case "r7l":
        cant = miSistema.listaInstancias[5].stock;
        pos=5;
        break;
    case "i7m":
        cant = miSistema.listaInstancias[6].stock;
        pos=6;
        break;
    case "i7l":
        cant = miSistema.listaInstancias[7].stock;
        pos=7;
        break;
    }
    // Mostrar el valor actual de 'cant' en el campo de entrada
    return pos;
}

function sumarStock() {
    let mensaje="Stock actualizado con éxito"
    let pos = modificarStock("selectStock");
    miSistema.listaInstancias[pos].stock++; // Incrementar la cantidad en
miSistema.listaInstancias
    document.querySelector("#cantStock").value =
miSistema.listaInstancias[pos].stock;
    document.querySelector("#msjStock").innerHTML= mensaje;
}

function restarStock() {
    let mensaje="";
    let pos = modificarStock("selectStock");
    if (miSistema.listaInstancias[pos].stock >
miSistema.listaInstancias[pos].cantidadAlquiladas) {
        miSistema.listaInstancias[pos].stock--;
        mensaje= "Stock actualizado con éxito" // Decrementar la cantidad en
miSistema.listaInstancias si es mayor que las maquinas de alquiler
        document.querySelector("#cantStock").value =
miSistema.listaInstancias[pos].stock;
    }
    else{
        mensaje="No es posible realizar la modificación de Stock, no puede
tener menos stock que máquinas en alquiler."
    }
    document.querySelector("#msjStock").innerHTML= mensaje;
}

```

```

// FIN GESTION DE STOCK

//ALQUILER DE MAQUINA
function mostrarAlquilerMaquina(){
    ocultarFormularios();
    document.querySelector("#mensajeAlquiler").innerHTML="";
    document.querySelector("#menuUsuario").style.display= "block";
    document.querySelector("#selectInstancias").style.display= "block";
}
function alquilerMaquina() {
    let mensaje=""
    let pos= modificarStock("tipoInstancia");
    if (miSistema.listaInstancias[pos].stock -
miSistema.listaInstancias[pos].cantidadAlquiladas >= 1) {
        miSistema.agregarAlquiler(miSistema.listaInstancias[pos]);
        mensaje= "Alquiler exitoso!"
    } else {
        mensaje="No fue posible realizar el alquiler."
    }
    document.querySelector("#mensajeAlquiler").innerHTML= mensaje;
}
function irATablaAlquiler(){
    ocultarFormularios();
    document.querySelector("#tablaInstanciasFiltro").innerHTML="";
    document.querySelector("#textoInstancias").innerHTML="";
    document.querySelector("#filtroApagadoEncendido").selectedIndex=0;
    document.querySelector("#menuUsuario").style.display= "block";
    document.querySelector("#maquinasAlquiladas").style.display= "block";
}
function tablaAlquiler(){
    let filtro= document.querySelector("#filtroApagadoEncendido").value;
    filtroTablaMaquinasAlquiladas(filtro)
}
function filtroTablaMaquinasAlquiladas(unFiltro) {
    let mensaje="";
    let lista= miSistema.generarListaFiltro(unFiltro,
miSistema.usuarioLogueado)

    if (lista.length>0){
        document.querySelector("#tablaInstanciasFiltro").style.display=
"block";
        let miTabla= document.querySelector("#tablaInstanciasFiltro");
        miTabla.innerHTML= "";
        let tablaHTML="";
        tablaHTML +=`<thead class="heading">
            <tr>
                <th>Instancia</th> <th>Estado</th> <th>Veces Iniciada</th>
<th>Acción</th>

```

```

        </tr>
    </thead>`
    mensaje = "";
    for (let pos = 0; pos < lista.length; pos++) {
        let objAlquiler= lista[pos];
        tablaHTML+=`<tr>
            <td>${objAlquiler.instancia.nombre}</td>
            <td>${objAlquiler.estado}</td> <td>${objAlquiler.contadorEncendido}</td>
            <td><input type="button" id="${objAlquiler.idAlquiler}"
value="Cambiar"></td>`
        }
        miTabla.innerHTML+= tablaHTML;

        for (let pos=0;pos<lista.length;pos++){
            let objAlquiler = lista[pos];
            document.querySelector("#"+
objAlquiler.idAlquiler).addEventListener("click", cambiar);
        }

    } else {
        document.querySelector("#tablaInstanciasFiltro").style.display=
"none";
        mensaje="No hay alquileres."
    }
    document.querySelector("#textoInstancias").innerHTML = mensaje;
}
function cambiar(){
    let idAlquiler = this.id;
    miSistema.cambiarEstadoMaquina(idAlquiler);
    tablaAlquiler();
}
// FIN ALQUILER

//GASTOS
function irATablaGastos(){
    ocultarFormularios();
    document.querySelector("#menuUsuario").style.display= "block";
    document.querySelector("#costosAlquiler").style.display= "block";
    document.querySelector("#textoCostos").innerHTML=""
    let tabla= miSistema.alquileresAgrupados()
    if (tabla==""){
        document.querySelector("#textoCostos").innerHTML = "No tienes alquileres
aún."
    }
    document.querySelector("#tablaCostos").innerHTML=tabla;
}

```



```

//GANANCIAS
function irATablaGanancias(){
    ocultarFormularios();
    document.querySelector("#menuAdmin").style.display = "block";
    document.querySelector("#informeDeMaquinas").style.display = "block";
    let tabla= miSistema.generarGanancias()
    let mensaje="";

    document.querySelector("#tablaInformeMaquinas").innerHTML=tabla;
    if(tabla == ""){
        mensaje = "No tienes alquileres aún.";
    }else{
        mensaje="El total de ingresos es: U$S " + miSistema.totalIngresos;
    }
    document.querySelector("#textoInformeMaquinas").innerHTML = mensaje;
}

// Acceso al sistema
let miSistema = new Sistema ();

```

2.3 *clases.js*

```
class Sistema {
  constructor(){
    this.listaUsuarios = new Array();
    this.listaAlquiler = new Array();
    this.usuarioLogueado= null; //va a guardar un objeto no asignado aún.
    this.listaInstancias= new Array();
    this.ultimoIdUsuario=0;
    this.ultimoAlquiler=0;
    this.totalIngresos=0;
  }

  precargarDatos(){
    // Precargo usuarios
    this.agregarUsuario("Admin1", "Admin1", "Fernando", "Perez", "1234-1234-1234-1234"); //admin
    this.listaUsuarios[0].estado="activo"
    this.listaUsuarios[0].tipo="admin"

    this.agregarUsuario("Admin2", "Admin2", "Lorena", "Martinez", "1234-1234-1234-1234"); //admin
    this.listaUsuarios[1].estado="activo";
    this.listaUsuarios[1].tipo="admin";

    this.agregarUsuario("Admin3", "Admin3", "Facundo", "Ibañez", "1234-1234-1234-1234"); //admin
    this.listaUsuarios[2].estado="activo";
    this.listaUsuarios[2].tipo="admin";

    this.agregarUsuario("Admin4", "Admin4", "Lucía", "Romero", "1234-1234-1234-1234"); //admin
    this.listaUsuarios[3].estado="activo";
    this.listaUsuarios[3].tipo="admin";

    this.agregarUsuario("Admin5", "Admin5", "Julio", "Canzani", "1234-1234-1234-1234"); //admin
    this.listaUsuarios[4].estado="activo";
    this.listaUsuarios[4].tipo="admin";

    this.agregarUsuario("usu1","Usu1111","María","Gonzalez", "1234-1234-1234-1234"); //usuario común
    this.listaUsuarios[5].estado="activo";

    this.agregarUsuario("usu2","Usu2222","Mario", "Castro", "1234-1234-1234-1234"); //usuario común
    this.listaUsuarios[6].estado="activo";
```

```

        this.agregarUsuario("usu3","Usu3333","Pedro", "Alvarez", "1234-1234-1234-1234");//usuario común
        this.listaUsuarios[7].estado="activo";

        this.agregarUsuario("usu4","Usu4444","Paula", "Perez", "1234-1234-1234-1234");//usuario común
        this.listaUsuarios[8].estado="bloqueado";

        this.agregarUsuario("usu5","Usu5555","Florencia", "Castillos", "1234-1234-1234-1234");//usuario común
        this.listaUsuarios[9].estado="pendiente"

//precarga de instancias
this.agregarInstancia("c7.small", "0", "20", "2.50");
this.agregarInstancia("c7.medium", "9", "30", "3.50");
this.agregarInstancia("c7.large", "10", "50", "6");
this.agregarInstancia("r7.small", "10", "35", "4");
this.agregarInstancia("r7.medium", "18", "50", "6.50");
this.agregarInstancia("r7.large", "9", "60", "7");
this.agregarInstancia("i7.medium", "15", "30", "3.50");
this.agregarInstancia("i7.large", "16", "50", "6.50");

//precarga de alquileres
this.agregarAlquiler( this.listaInstancias[2]);
this.listaAlquiler[0].usuario= this.listaUsuarios[5];

this.agregarAlquiler( this.listaInstancias[2]);
this.listaAlquiler[1].usuario= this.listaUsuarios[5];

this.agregarAlquiler( this.listaInstancias[1]);
this.listaAlquiler[2].usuario= this.listaUsuarios[5];

this.agregarAlquiler( this.listaInstancias[1]);
this.listaAlquiler[3].usuario= this.listaUsuarios[7];

this.agregarAlquiler( this.listaInstancias[3]);
this.listaAlquiler[4].usuario= this.listaUsuarios[7];

this.agregarAlquiler( this.listaInstancias[4]);
this.listaAlquiler[5].usuario= this.listaUsuarios[7];

this.agregarAlquiler( this.listaInstancias[4]);
this.listaAlquiler[6].usuario= this.listaUsuarios[9];

this.agregarAlquiler( this.listaInstancias[7]);
this.listaAlquiler[7].usuario= this.listaUsuarios[9];

this.agregarAlquiler( this.listaInstancias[7]);

```

```

        this.listaAlquiler[8].usuario= this.listaUsuarios[9];

        this.agregarAlquiler( this.listaInstancias[6]);
        this.listaAlquiler[9].usuario= this.listaUsuarios[9];

    }

    // PUSHES
    agregarUsuario(unNombreUsuario, unaContraseña, unNombre, unApellido,
unaTarjeta, unTipo, unEstado){
        let esta = this.verificarNombreUsuarioRepetido(unNombreUsuario);
        if (!esta){
            let objUsuario = new Usuario(unNombreUsuario, unaContraseña,
unNombre, unApellido, unaTarjeta, unTipo, unEstado);
            this.listaUsuarios.push(objUsuario);

        }
        return !esta;
    }
    agregarInstancia(unNombre, stock, unPrecioAlquiler, unPrecioEncendido){
        let objInstancia = new TipoInstancia(unNombre, stock,
unPrecioAlquiler, unPrecioEncendido);
        this.listaInstancias.push(objInstancia);
    }
    agregarAlquiler(unObjInstancia){
        let objAlquiler = new Alquiler(unObjInstancia, this.usuarioLogueado,
this.ultimoAlquiler);
        this.listaAlquiler.push(objAlquiler);
        this.ultimoAlquiler++;
        this.sumarCantidadAlquiladas(unObjInstancia);

    }
    sumarCantidadAlquiladas(objInstancia){
        for (let pos=0; pos< this.listaInstancias.length; pos++){
            if (this.listaInstancias[pos].nombre == objInstancia.nombre){
                this.listaInstancias[pos].cantidadAlquiladas ++;
            }
        }
    }
    restarCantidadAlquiladas(objInstancia){
        for (let pos=0; pos< this.listaInstancias.length; pos++){
            if (this.listaInstancias[pos].nombre == objInstancia.nombre){
                this.listaInstancias[pos].cantidadAlquiladas --;
            }
        }
    }

}

// VERIFICACIONES DEL REGISTRO

```

```

verificarNombreyApellido (unNombre, unApellido){
    let nombreApellidoValido= true;

    for (let pos=0; pos<unNombre.length ; pos++ ){
        let letra = unNombre.charAt(pos);
        if(letra.charCodeAt(0)!= 32 && letra.charCodeAt(0)<65 &&
letra.charCodeAt(0)>90 && letra.charCodeAt(0)<97 && letra.charCodeAt(0)>122){
            nombreApellidoValido=false;
        }
    }
    for (let pos=0; pos<unApellido.length ; pos++ ){
        let letra = unApellido.charAt(pos);
        if(letra.charCodeAt(0)!= 32 && letra.charCodeAt(0)<65 &&
letra.charCodeAt(0)>90 && letra.charCodeAt(0)<97 && letra.charCodeAt(0)>122){
            nombreApellidoValido=false;
        }
    }
    return nombreApellidoValido
}

verificarNombreUsuario (unUsuario){
    let nombreUsuarioValido=true;
    let contadorNumero=0;
    let contadorLetra=0;
    for (let pos=0; pos<unUsuario.length ; pos++ ){
        let caracter = unUsuario.charAt(pos);
        if(!isNaN (caracter)){ //es un número
            contadorNumero++;
        }
        if (caracter.charCodeAt(0)>=65 && caracter.charCodeAt(0)<=90 ||
caracter.charCodeAt(0)>=97 && caracter.charCodeAt(0)<=122){
            contadorLetra++;
        }
    }
    if(contadorLetra==0 || contadorNumero==0){
        nombreUsuarioValido=false;
    }

    return nombreUsuarioValido;
}

verificarContraseñaValida(unaContraseña){
    let contraseñaValida=false;
    let contadorMin = 0;
    let contadorMay = 0;
    let contadorNumero = 0;
    let espacios=0;
    for(let pos=0; pos< unaContraseña.length; pos++){

```

```

        let letra=unaContraseña.charAt(pos)
        if(letra== " "){
            espacios++;
        }
        if(letra.charCodeAt(0)>=65 && letra.charCodeAt(0)<=90){
            contadorMay++;
        }
        if(letra.charCodeAt(0)>=97 && letra.charCodeAt(0)<=122){
            contadorMin++;
        }
        if(letra.charCodeAt(0)>=48 && letra.charCodeAt(0)<=57){
            contadorNumero++;
        }
    }
    if(contadorMay>0 && contadorMin>0 && contadorNumero>0 &&
espacios==0){
        contraseñaValida=true;
    }
    return contraseñaValida;
}

//VERIFICACIÓN DE TARJETA
algoritmoLuhn(pNumero) {
    /*Se estara iterando numero a numero, desde el final del string hasta el
primer caracter, se estarán
sumando y sustituyendo por duplicado cuando sea par, ya que sería el
segundo nro. */
    let suma = 0;
    let digitoVerificadorX = Number(pNumero.charAt(pNumero.length - 1));
    let contador = 0; //para saber cuando estamos en los segundos, lo pares.
    let haynro = true;
    let i = pNumero.length - 2; //el penúltimo.

    //Mientras que esté parado en una posición valida
    while (i >= 0 && haynro) {
        //Obtener el numero
        let caracter = pNumero.charAt(i);
        //Valida que el número sea válido
        if (!isNaN(caracter)) {
            let num = Number(caracter);
            //Duplicando cada segundo dígito
            if (contador % 2 == 0) {
                num = this.duplicarPar(num); //porque si es mayor a 9 se deben
sumar.
            }
            suma += num;
        } else {
            haynro = false;

```

```

    }
    i--;
    contador++;
}
let digitoVerificadorValido = this.checkDigito(suma, digitoVerificadorX);
let modulodelasumaValido = this.checkModulo(suma, digitoVerificadorX);
return digitoVerificadorValido && modulodelasumaValido;

}

duplicarPar(pNum) {
  pNum = pNum * 2;
  if (pNum > 9) {
    /*Si el resultado del multiplicación es mayor a 9 entonces lo
descomponemos y sumamos.
Como el numero sera x>=10 && x<=19
Entonces es 1+(num % 10) 1 más el resto de dividir entre 10.*/
    pNum = 1 + (pNum % 10);
  }
  return pNum;
}

checkDigito(pSuma, pDigito) {
  /* 1. Calcular la suma de los dígitos (67).
  2. Multiplicar por 9 (603).
  3. Tomar el último dígito (3).
  4. El resultado es el dígito de chequeo.*/
  let total = 9 * pSuma;
  let ultimoNro = total % 10
  return ultimoNro === pDigito;
}

checkModulo(pSuma, pDigito) {
  /*
  Si el total del módulo 10 es igual a 0 (si el total termina en cero),
entonces el número es válido
de acuerdo con la fórmula Luhn, de lo contrario no es válido.
*/
  let total = pSuma + pDigito;
  let validacionFinal = false;
  if (total % 10 === 0 && total !== 0) {
    validacionFinal = true;
  }
  return validacionFinal;
}
//FIN VERIFICACIÓN TARJETA

verificarNombreUsuarioRepetido(unNombreUsuario){
  let existe = false;

```

```

        for (let pos=0;pos<this.listaUsuarios.length && !existe; pos++){
            let objUsuario = this.listaUsuarios[pos]
            if (objUsuario.nombreUsuario==unNombreUsuario){
                existe = true;
            }
        }
        return existe;
    }

    //LOGIN
    revisarUsuarioContraseña (unNombreUsuario, unaContraseña){
        let res = "incorrecto";
        let seguir = true;
        for (let pos = 0; pos < this.listaUsuarios.length && seguir; pos++) {
            let unUsuario = this.listaUsuarios[pos];
            if(unUsuario.nombreUsuario.toUpperCase()==
unNombreUsuario.toUpperCase() && unUsuario.contraseña== unaContraseña ){
                if(unUsuario.estado=="activo"){
                    res= unUsuario.tipo;
                    this.usuarioLogueado= unUsuario;
                } else{
                    res= "Usuario no habilitado"
                }
                seguir = false;
            }
        }
        return res;
    }

    // TABLA DE USUARIOS Y ESTADOS
    obtenerUsuarioComun(){
        let resultado= new Array();
        for (let pos=0; pos<this.listaUsuarios.length; pos++){
            let objUsuario= this.listaUsuarios[pos];
            if(objUsuario.tipo=="usuario"){
                resultado.push(objUsuario);
            }
        }
        return resultado;
    }

    cambiarEstado(idUsuario){
        for (let pos=0; pos<this.listaUsuarios.length; pos++){
            let objUsuario= this.listaUsuarios[pos];
            if(objUsuario.nombreUsuario== idUsuario){
                if (objUsuario.estado=="activo"){
                    objUsuario.estado= "bloqueado";
                    this.eliminarAlquileres(objUsuario);

                } else {

```



```

        if(objUsuario.estado== "pendiente"){
            objUsuario.estado="activo";

        } else {
            if(objUsuario.estado== "bloqueado"){
                objUsuario.estado="activo";

            }
        }
    }
}

eliminarAlquileres(unUsuario){

    for (let n = this.listaAlquiler.length - 1; n >= 0; n--){
        if(this.listaAlquiler[n].usuario.nombreUsuario==
unUsuario.nombreUsuario){
            this.listaAlquiler.splice(n,1); // elimino el alquiler del
usuario que acabo de bloquear
            this.restarCantidadAlquiladas(this.listaAlquiler[n].instancia)
;
        }
    }
}

// FIN TABLA DE USUARIOS Y ESTADOS

// TABLA DE MAQUINAS ALQUILADAS Y ESTADOS

generarListaFiltro(unFiltro, unUsuario){
    if (unFiltro== "T") { // genera lista de todos los alquileres
        let listaAlquilerTodos = new Array();
        for (let pos=0;pos<this.listaAlquiler.length;pos++){
            let objAlquiler = this.listaAlquiler[pos];
            if (objAlquiler.usuario==unUsuario){
                listaAlquilerTodos.push(objAlquiler);
            }
        }
        return listaAlquilerTodos;
    }
    if(unFiltro== "A"){ // genera lista unicamente con las instancias
apagadas
        let listaAlquilerApagado= new Array();
        for (let pos=0;pos<this.listaAlquiler.length;pos++){
            let objAlquiler = this.listaAlquiler[pos];

```

```

        if (objAlquiler.usuario==unUsuario && objAlquiler.estado==
"apagado"){
            listaAlquilerApagado.push(objAlquiler);
        }
    }
    return listaAlquilerApagado;
}
if(unFiltro== "E"){ // genera lista unicamente con las instancias
encendidas
    let listaAlquilerEncendido = new Array();
    for (let pos=0;pos<this.listaAlquiler.length;pos++){
        let objAlquiler = this.listaAlquiler[pos];
        if (objAlquiler.usuario==unUsuario && objAlquiler.estado==
"encendido"){
            listaAlquilerEncendido.push(objAlquiler);
        }
    }
    return listaAlquilerEncendido;
}
}
cambiarEstadoMaquina(idAlquiler){
    for (let pos=0; pos<this.listaAlquiler.length; pos++){
        let objAlquiler= this.listaAlquiler[pos];
        if(objAlquiler.idAlquiler== idAlquiler){
            if (objAlquiler.estado=="encendido"){
                objAlquiler.estado= "apagado";

            } else {
                objAlquiler.estado="encendido";
                objAlquiler.contadorEncendido++;
            }
        }
    }
}

}

// FIN TABLAS DE MAQUINAS ALQUILADAS Y ESTADOS

// INFO TABLA GASTOS
alquileresAgrupados(){
    let tablaHTML="";
    let lista= this.generarListaFiltro("T", this.usuarioLogueado);
    if(lista.length>0){
        tablaHTML = `<thead class="heading"><tr><th>Tipo de
instancia</th><th>Costo por encendido</th><th>Veces encendidas</th><th>Gasto
total</th></tr></thead>`
        for (let i = 0; i <this.listaInstancias.length; i++) {
            let objInstancia= this.listaInstancias[i]
            let contadorTotalEncendidos = 0;

```

```

        let contadorTotalAlquileres = 0;
        for (let pos = 0; pos < lista.length; pos++) {
            let objAlquiler= lista[pos];
            if (objAlquiler.instancia==objInstancia) {
                contadorTotalEncendidos+= objAlquiler.contadorEncendido;
                contadorTotalAlquileres++;
            }
        }
        if (contadorTotalAlquileres > 0) {
            tablaHTML+=`<tr><td>${objInstancia.nombre}</td>
                <td>U$S ${objInstancia.encendido}</td>
                <td> ${contadorTotalEncendidos}</td>
                <td>U$S ${ (contadorTotalEncendidos *
objInstancia.encendido) + (objInstancia.alquiler*contadorTotalAlquileres)}<td>
                </tr>`
        }
    }
    }
    return tablaHTML;
}

//INFORME DE MAQUINAS

generarGanancias(){
    let tablaHTML="";
    this.totalIngresos=0;
    if(this.listaAlquiler.length>0){ // Si existen alquileres
        tablaHTML = `<thead class="heading"><tr><th>Tipo de
instancia</th><th>Cantidad en Alquiler</th><th>Ingreso Total
Actual</th></tr></thead>`
        for (let i = 0; i <this.listaInstancias.length; i++) {
            let objInstancia= this.listaInstancias[i]
            let contadorTotalEncendidos = 0;

            for (let pos = 0; pos < this.listaAlquiler.length; pos++) {
                let objAlquiler= this.listaAlquiler[pos];
                if (objAlquiler.instancia==objInstancia) {
                    contadorTotalEncendidos+= objAlquiler.contadorEncendido;
                }
            }
            if (objInstancia.cantidadAlquiladas > 0) {
                tablaHTML+=`<tr><td>${objInstancia.nombre}</td>
                    <td>${objInstancia.cantidadAlquiladas} </td>
                    <td>U$S ${ (contadorTotalEncendidos *
objInstancia.encendido) +
(objInstancia.alquiler*objInstancia.cantidadAlquiladas)}<td></tr>`
                this.totalIngresos+=(contadorTotalEncendidos *
objInstancia.encendido) +
(objInstancia.alquiler*objInstancia.cantidadAlquiladas);
            }
        }
    }
}

```

```

        }
    }
    }
    return tablaHTML;
}
}

class Usuario{
    constructor(unNombreUsuario, unaContraseña, unNombre, unApellido,
unaTarjeta){
        this.nombre = unNombre;
        this.apellido = unApellido;
        this.nombreUsuario= unNombreUsuario;
        this.contraseña = unaContraseña;
        this.tarjetaDeCredito = unaTarjeta;
        this.tipo = "usuario";
        this.estado= "pendiente";
        this.id= "USUARIO_ID_" + miSistema.ultimoIdUsuario;
        miSistema.ultimoIdUsuario++;
    }
}

class Alquiler{
    constructor(unObjInstancia, unObjUsuario, unIdAlquiler){
        this.instancia = unObjInstancia;
        this.usuario = unObjUsuario;
        this.estado = "encendido";
        this.contadorEncendido = 0;
        this.idAlquiler = "INSTANCE_ID_" + unIdAlquiler;
    }
}

class TipoInstancia {
    constructor(unNombre, unaCantidad, unPrecioAlquiler, unPrecioEncendido){
        this.nombre= unNombre;
        this.stock= unaCantidad;
        this.cantidadAlquiladas = 0;
        this.alquiler=unPrecioAlquiler;
        this.encendido= unPrecioEncendido;
    }
}

```

2.4 estilos.css

```
@charset "utf-8";
/* CSS Document */

@import
url("https://fonts.googleapis.com/css2?family=Roboto+Flex:opsz@8..144&display=
swap");

.clearfix:after {
  visibility: hidden;
  display: block;
  font-size: 0;
  content: " ";
  clear: both;
  height: 0;
}

.clearfix {
  display: inline-block;
}
/* start commented backslash hack */

* html .clearfix {
  height: 1%;
}

.clearfix {
  display: block;
}
/* close commented backslash hack */

body {
  font-family: "Roboto Flex", sans-serif;
  background-color: fafafa;
  margin: 0;
}

/* Estilos contenedor de todo el documento */

#contenedor {
  width: 100%;
  height: 100vh;
  background-color: #fafafa;
}

/* Estilos contenedor de todo el contenido */

#contenido {
```

```

padding: 20px;
float: left;
}

/* Titulos */

h1,
h2,
h3,
h5,
h6 {
font-family: "Roboto Flex", sans-serif;
color: #002e4a;
}

h2 {
font-weight: 800;
color: #c0ca33;
text-transform: uppercase;
}

h4 {
margin-top: 26px;
font-weight: 600;
color: #f4f5ed;
}

/* Estilos NAV */

#aux a {
color: #002e4a;
}

#menuUsuario {
float: right;
margin-top: 5px;
margin-right: 10px;
max-width: 400px;
}

#menuAdmin {
float: right;
margin-top: 5px;
margin-right: 10px;
max-width: 400px;
}

#menuUsuario li {
font-family: "Roboto Flex", sans-serif;
list-style-type: none;
padding: 10px;
margin: 5px;

```

```

float: left;
}
#menuAdmin li {
  font-family: "Roboto Flex", sans-serif;
  list-style-type: none;
  padding: 10px;
  margin: 5px;
  float: left;
}

#logo {
  padding: 20px;
  float: left;
  width: 60px;
  height: auto;
  float: left;
}

#aux {
  float: right;
  clear: right;
  padding: 0 24px 10px 0;
}

#menuUsuario a {
  cursor: pointer;
  color: #002e4a;
  font-size: 14px;
  font-weight: 600;
}

#menuAdmin a {
  cursor: pointer;
  color: #002e4a;
  font-size: 14px;
  font-weight: 600;
}

#aux a {
  text-decoration: underline;
  cursor: pointer;
  color: #002e4a;
  font-size: 16px;
}

#menuUsuario a:hover {
  color: #c0ca33;
}

#menuAdmin a:hover {
  color: #c0ca33;
}

```

```

}

#cabezal {
  background-color: #f5f5f7;
  clear: both;
  color: fafafa;
  text-transform: uppercase;
}

/* Estilos tabla */
table,
th,
td {
  border-collapse: collapse;
  border: none;
  padding-left: 15px;
  padding-bottom: 5px;
  padding-top: 5px;
  background-color: #fafafa;
}
.heading {
  border-bottom: 1px solid black;
  background-color: #fafafa;
  color: #002e4a;
  font-weight: normal;
}
table img,
td > input[type="button"] {
  width: 100px;
  min-width: 50px;
}

table tr:nth-child(odd) td {
  background-color: #ececfc;
}
table tr:nth-child(even) td {
  background-color: #fafafa;
}

/* Estilos label de los inputs */
label {
  margin-left: 11px;
  text-transform: uppercase;
  font-size: 13px;
}

p.ver-mas {
  text-align: center;
  cursor: pointer;
}

```



```

    font-weight: 700;
    color: #fc5130;
}

/* Estilos de los inputs */

/* input largo */
input[type="text"], [type="number"], [type="password"] {
    margin: 0 10px 10px 0;
    display: block;
    width: 60%;
    padding: 0.375rem 0.75rem;
    font-size: 1rem;
    line-height: 1.5;
    color: #495057;
    background-color: fafafa;
    background-clip: padding-box;
    border: 1.5px solid #4a5a70;
    border-radius: 0.25rem;
    transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
}

/* select */
select {
    margin: 0 10px 10px 0;
    display: block;
    width: 100%;
    padding: 0.375rem 0.75rem;
    font-size: 1rem;
    line-height: 1.5;
    color: #495057;
    background-color: fafafa;
    background-clip: padding-box;
    border: 1.5px solid #4a5a70;
    border-radius: 0.25rem;
    transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
}

/* area de texto */
.textArea {
    margin: 0 10px 10px 0;
    display: block;
    width: 30%;
    min-height: 80px;
    color: #495057;
    background-color: fafafa;
    background-clip: padding-box;
    border: 1.5px solid #4a5a70;
    border-radius: 0.25rem;
}

```

```

}

/* boton */
input[type="button"], button, .button {
  background-color: #002e4a;
  color: #fafafa;
  border: 1px solid rgb(0, 58, 117);
  border-radius: 6px;
  min-width: 74px;
  height: 35px;
  padding: 10px;
  text-transform: uppercase;
}

input[type="button"]:hover {
  background-color: #c0ca33;
  border: 1px solid #eefe00;
  color: #fafafa;
  box-shadow: none;
  border-radius: 6px;
  min-width: 74px;
  height: 35px;
  padding: 10px;
  text-transform: uppercase;
}

input[type="button"]:disabled {
  background-color: lightgray;
}

/* Estilos de los parrafos del contenido */

#lateral {
  background-color: #efefef;
  width: 290px;
  float: left;
  padding: 20px;
}

#lateral,
#contenido {
  min-height: 250px;
}

.articles2col {
  display: flex;
  align-items: self-start;
  margin: 15px 0 15px 0;
}

.articles1col {

```

```

display: flex;
flex-flow: column;
width: 70%;
margin: 10 auto;
}
.subContent {
display: flex;
justify-content: center;
flex-flow: column;
align-items: center;
}

/* Estilos del footer */

#footer {
background-color: #002e4a;
clear: both;
padding: 20px;
color: #fafafa;
font-size: 14px;
font-weight: 600;
text-align: center;
height: auto;
}
.footerNombres, .footerDos, .footerTres{
color: #fafafa;
}
#footer a {
cursor: pointer;
color: #fafafa;
font-size: 14px;
font-weight: 600;
text-decoration: none;
text-align: center;
}
#footer h4,
#footer p {
margin-bottom: 0;
}

```

3 LISTADO DE INFORMACIÓN PRECARGADA

> miSistema.listaUsuarios

< ▼ (10) [Usuario, Usuario, Usuario, Usuario, Usuario, Usuario, Usuario, Usuario, Usuario, Usuario] ⓘ

- ▼ 0: Usuario
 - apellido: "Perez"
 - contraseña: "Admin1"
 - estado: "activo"
 - id: "USUARIO_ID_0"
 - nombre: "Fernando"
 - nombreUsuario: "Admin1"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "admin"
 - ▶ [[Prototype]]: Object
- ▼ 1: Usuario
 - apellido: "Martínez"
 - contraseña: "Admin2"
 - estado: "activo"
 - id: "USUARIO_ID_1"
 - nombre: "Lorena"
 - nombreUsuario: "Admin2"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "admin"
 - ▶ [[Prototype]]: Object
- ▼ 2: Usuario
 - apellido: "Ibañez"
 - contraseña: "Admin3"
 - estado: "activo"
 - id: "USUARIO_ID_2"
 - nombre: "Facundo"
 - nombreUsuario: "Admin3"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "admin"
 - ▶ [[Prototype]]: Object
- ▼ 3: Usuario
 - apellido: "Romero"
 - contraseña: "Admin4"
 - estado: "activo"
 - id: "USUARIO_ID_3"
 - nombre: "Lucía"
 - nombreUsuario: "Admin4"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "admin"
 - ▶ [[Prototype]]: Object
- ▼ 4: Usuario
 - apellido: "Canzani"
 - contraseña: "Admin5"
 - estado: "activo"
 - id: "USUARIO_ID_4"
 - nombre: "Julio"
 - nombreUsuario: "Admin5"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "admin"
 - ▶ [[Prototype]]: Object
- ▼ 5: Usuario
 - apellido: "Gonzalez"
 - contraseña: "Usu1111"
 - estado: "activo"
 - id: "USUARIO_ID_5"
 - nombre: "María"
 - nombreUsuario: "usu1"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
- ▼ 6: Usuario
 - apellido: "Castro"
 - contraseña: "Usu2222"
 - estado: "activo"
 - id: "USUARIO_ID_6"
 - nombre: "Mario"
 - nombreUsuario: "usu2"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
- ▼ 7: Usuario
 - apellido: "Alvarez"
 - contraseña: "Usu3333"
 - estado: "activo"
 - id: "USUARIO_ID_7"
 - nombre: "Pedro"
 - nombreUsuario: "usu3"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
- ▼ 8: Usuario
 - apellido: "Perez"
 - contraseña: "Usu4444"
 - estado: "bloqueado"
 - id: "USUARIO_ID_8"
 - nombre: "Paula"
 - nombreUsuario: "usu4"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
- ▼ 9: Usuario
 - apellido: "Castillos"
 - contraseña: "Usu5555"
 - estado: "pendiente"
 - id: "USUARIO_ID_9"
 - nombre: "Florencia"
 - nombreUsuario: "usu5"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
 - length: 10
 - ▶ [[Prototype]]: Array(0)

> miSistema.listaAlquiler

< ▼ (10) [Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler] ⓘ

- ▼ 0: Alquiler
 - contadorEncendido: 0
 - estado: "encendido"
 - idAlquiler: "INSTANCE_ID_0"
 - ▼ instancia: TipoInstancia
 - alquiler: "50"
 - cantidadAlquiladas: 2
 - encendido: "6"
 - nombre: "c7.large"
 - stock: "10"
 - ▶ [[Prototype]]: Object
 - ▼ usuario: Usuario
 - apellido: "Gonzalez"
 - contraseña: "Usu1111"
 - estado: "activo"
 - id: "USUARIO_ID_5"
 - nombre: "María"
 - nombreUsuario: "usu1"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
 - ▶ [[Prototype]]: Object
- ▼ 1: Alquiler
 - contadorEncendido: 0
 - estado: "encendido"
 - idAlquiler: "INSTANCE_ID_1"
 - ▼ instancia: TipoInstancia
 - alquiler: "50"
 - cantidadAlquiladas: 2
 - encendido: "6"
 - nombre: "c7.large"
 - stock: "10"
 - ▶ [[Prototype]]: Object
 - ▼ usuario: Usuario
 - apellido: "Gonzalez"
 - contraseña: "Usu1111"
 - estado: "activo"
 - id: "USUARIO_ID_5"
 - nombre: "María"
 - nombreUsuario: "usu1"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
 - ▶ [[Prototype]]: Object
- ▼ 2: Alquiler
 - contadorEncendido: 0
 - estado: "encendido"
 - idAlquiler: "INSTANCE_ID_2"
 - ▼ instancia: TipoInstancia
 - alquiler: "30"
 - cantidadAlquiladas: 2
 - encendido: "3.50"
 - nombre: "c7.medium"
 - stock: "9"
 - ▶ [[Prototype]]: Object
 - ▼ usuario: Usuario
 - apellido: "Gonzalez"
 - contraseña: "Usu1111"
 - estado: "activo"
 - id: "USUARIO_ID_5"
 - nombre: "María"
 - nombreUsuario: "usu1"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
 - ▶ [[Prototype]]: Object
- ▼ 3: Alquiler
 - contadorEncendido: 0
 - estado: "encendido"
 - idAlquiler: "INSTANCE_ID_3"
 - ▼ instancia: TipoInstancia
 - alquiler: "30"
 - cantidadAlquiladas: 2
 - encendido: "3.50"
 - nombre: "c7.medium"
 - stock: "9"
 - ▶ [[Prototype]]: Object
 - ▼ usuario: Usuario
 - apellido: "Alvarez"
 - contraseña: "Usu3333"
 - estado: "activo"
 - id: "USUARIO_ID_7"
 - nombre: "Pedro"
 - nombreUsuario: "usu3"
 - tarjetaDeCredito: "1234-1234-1234-1234"
 - tipo: "usuario"
 - ▶ [[Prototype]]: Object
 - ▶ [[Prototype]]: Object

```
> miSistema.listaAlquiler

< ▼ (10) [Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler] ⓘ
▼ 4: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_4"
  ▼ instancia: TipoInstancia
    alquiler: "35"
    cantidadAlquiladas: 1
    encendido: "4"
    nombre: "r7.small"
    stock: "10"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Alvarez"
    contraseña: "Usu3333"
    estado: "activo"
    id: "USUARIO_ID_7"
    nombre: "Pedro"
    nombreUsuario: "usu3"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
▼ 5: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_5"
  ▼ instancia: TipoInstancia
    alquiler: "50"
    cantidadAlquiladas: 2
    encendido: "6.50"
    nombre: "r7.medium"
    stock: "18"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Alvarez"
    contraseña: "Usu3333"
    estado: "activo"
    id: "USUARIO_ID_7"
    nombre: "Pedro"
    nombreUsuario: "usu3"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
▼ 6: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_6"
  ▼ instancia: TipoInstancia
    alquiler: "50"
    cantidadAlquiladas: 2
    encendido: "6.50"
    nombre: "r7.medium"
    stock: "18"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Castillos"
    contraseña: "Usu5555"
    estado: "pendiente"
    id: "USUARIO_ID_9"
    nombre: "Florencia"
    nombreUsuario: "usu5"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
▼ 7: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_7"
  ▼ instancia: TipoInstancia
    alquiler: "50"
    cantidadAlquiladas: 2
    encendido: "6.50"
    nombre: "i7.large"
    stock: "16"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Castillos"
    contraseña: "Usu5555"
    estado: "pendiente"
    id: "USUARIO_ID_9"
    nombre: "Florencia"
    nombreUsuario: "usu5"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
```

```
> miSistema.listaAlquiler

< ▼ (10) [Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler, Alquiler] ⓘ
▼ 8: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_8"
  ▼ instancia: TipoInstancia
    alquiler: "50"
    cantidadAlquiladas: 2
    encendido: "6.50"
    nombre: "i7.large"
    stock: "16"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Castillos"
    contraseña: "Usu5555"
    estado: "pendiente"
    id: "USUARIO_ID_9"
    nombre: "Florencia"
    nombreUsuario: "usu5"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
▼ 9: Alquiler
  contadorEncendido: 0
  estado: "encendido"
  idAlquiler: "INSTANCE_ID_9"
  ▼ instancia: TipoInstancia
    alquiler: "30"
    cantidadAlquiladas: 1
    encendido: "3.50"
    nombre: "i7.medium"
    stock: "15"
    ► [[Prototype]]: Object
  ▼ usuario: Usuario
    apellido: "Castillos"
    contraseña: "Usu5555"
    estado: "pendiente"
    id: "USUARIO_ID_9"
    nombre: "Florencia"
    nombreUsuario: "usu5"
    tarjetaDeCredito: "1234-1234-1234-1234"
    tipo: "usuario"
    ► [[Prototype]]: Object
  ► [[Prototype]]: Object
```

> miSistema.listaInstancias

< ▼ (8) [TipoInstancia, TipoInstancia, TipoInstancia, TipoInstancia, TipoInstancia, TipoInstancia, TipoInstancia, TipoInstancia] ⓘ

```
▼ 0: TipoInstancia
  alquiler: "20"
  cantidadAlquiladas: 0
  encendido: "2.50"
  nombre: "c7.small"
  stock: "0"
  ▶ [[Prototype]]: Object
▼ 1: TipoInstancia
  alquiler: "30"
  cantidadAlquiladas: 2
  encendido: "3.50"
  nombre: "c7.medium"
  stock: "9"
  ▶ [[Prototype]]: Object
▼ 2: TipoInstancia
  alquiler: "50"
  cantidadAlquiladas: 2
  encendido: "6"
  nombre: "c7.large"
  stock: "10"
  ▶ [[Prototype]]: Object
▼ 3: TipoInstancia
  alquiler: "35"
  cantidadAlquiladas: 1
  encendido: "4"
  nombre: "r7.small"
  stock: "10"
  ▶ [[Prototype]]: Object
▼ 4: TipoInstancia
  alquiler: "50"
  cantidadAlquiladas: 2
  encendido: "6.50"
  nombre: "r7.medium"
  stock: "18"
  ▶ [[Prototype]]: Object
  ~~~~~
▼ 5: TipoInstancia
  alquiler: "60"
  cantidadAlquiladas: 0
  encendido: "7"
  nombre: "r7.large"
  stock: "9"
  ▶ [[Prototype]]: Object
▼ 6: TipoInstancia
  alquiler: "30"
  cantidadAlquiladas: 1
  encendido: "3.50"
  nombre: "i7.medium"
  stock: "15"
  ▶ [[Prototype]]: Object
▼ 7: TipoInstancia
  alquiler: "50"
  cantidadAlquiladas: 2
  encendido: "6.50"
  nombre: "i7.large"
  stock: "16"
  ▶ [[Prototype]]: Object
length: 8
▶ [[Prototype]]: Array(0)
```