

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACION 1

DOCUMENTO DE ANÁLISIS



Agustin Torres – 225494



Emanuel Rodriguez – 306299

N1F

Docente: Mauro Nacimiento

Analista Tecnologías de la Información

Fecha de entrega del documento 25/05/2023

Índice

1.	Descripción general del problema a resolver	4
1.1.	Tipos de usuario del sistema	4
1.2.	Listado de funcionalidades	4
2.	Detalle de Funcionalidades	5
2.1	F01 – Registrarse en la aplicación	5
2.1.1	Acceso	5
2.1.2	Descripción	5
2.1.3	Interfaz de usuario	6
2.2	F02 – Ingresar en la aplicación	6
2.2.1	Acceso	6
2.2.2	Descripción	7
2.2.3	Interfaz de usuario	7
2.3	F03 – Ingresar datos de una persona Censada	7
2.3.1	Acceso	7
2.3.2	Descripción	8
2.3.3	Interfaz de Usuario	8
2.4	F04 - Consultar censos pre completados	9
2.4.1	Acceso	9
2.4.2	Descripción	9
2.4.3	Interfaz	9
2.5	F05 – Reasignar personas para validar	9
2.5.1	Acceso	9
2.5.2	descripción	10

2.5.3	Interfaz	10
2.6	- F06 – Visualizar información estadística	10
2.6.1	- Acceso	10
2.6.2	descripción	10
2.6.3	Interfaz	11
2.7	- F07 – Registro en la aplicación	11
2.7.1	Acceso	11
2.7.2	Descripción	11
2.7.3	Interfaz	12
2.8	- F08 – Búsqueda de usuario	12
2.8.1	Acceso	12
2.8.2	descripción	12
2.8.3	Interfaz	13
2.9	- F09 Asignación Censista	13
2.9.1	Acceso	13
2.9.2	descripción	14
2.9.3	Interfaz	14
2.10	- F10 - Eliminar sus datos para el censo	14
2.10.1	Acceso	14
2.10.2	descripción	14
2.10.3	interfaz	15
2.11	- F11 - Lista de censados	15
2.11.1	Acceso	15
2.11.2	descripción	15
2.11.3	interfaz	16
3.	Informe de Testing	17

4.	Código de desarrollo.....	26
4.1	Lista de Pre-Cargas de datos.....	26
4.2	Código HTML:	27
4.4	Código Sistema.js	32
4.5	Codigo.js.....	39
4.6	Código Clases.js	55
4.7	Código Estilos.css.....	56

1. Descripción general del problema a resolver

La aplicación surge a partir de la necesidad de nuestro cliente, Curso de Programación I de las carreras Analista Programador/Analista en Tecnologías de la Información, para aportar un sistema de gestión de toda la información recabada del censo de ciudadanos. El proyecto es desarrollado en HTML 5/JavaScript, se enfoca en mejorar la eficiencia en cuanto a datos digitalizados y productividad.

1.1. Tipos de usuario del sistema

- Censista
- INVITADO (persona).

1.2. Listado de funcionalidades

F01 – Registrarse en la aplicación – Usuario: Censista

F02 – Ingresar en la aplicación – Usuario: Censista

F03 – Ingresar datos de una persona Censada – Usuario: Censista

F04 – Consultar censos pre completados – Usuario: Censista

F05 – Reasignar persona para validar – Usuario: Censista

F06 – Visualizar información estadística – Usuario: Censista

F07 – Registro en la aplicación – Usuario: Invitado(persona)

F08 - Búsqueda de usuario - Usuario: Invitado(persona)

F09 - Asignación censista - Usuario: Invitado(persona)

F10 - Eliminar sus datos para el censo - Usuario: Invitado(persona)

F11- Lista de censados - Usuario: Invitado(persona)

2. Detalle de Funcionalidades

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

2.1 F01 – Registrarse en la aplicación

2.1.1 Acceso

Censista

2.1.2 Descripción

El Censista podrá registrarse para poder ingresar en la aplicación y hacer uso de las distintas funcionalidades.

2.1.3 Interfaz de usuario



Registro

Nombre

Nombre Usuario

Contraseña

Confirmar Registro

Validaciones

Todos los datos para ingresar son obligatorios, el nombre de usuario no se podrá repetir, en caso de que se repita saldrá mensaje de error. LA contraseña deberá contar con un mínimo de 5 caracteres contado con al menos una mayúscula y un número.

2.2 F02 – Ingresar en la aplicación

2.2.1 Acceso

Censista

2.2.2 Descripción

El Censista una vez registrado o teniendo los datos precargados, podrá ingresar en la aplicación ingresando sus datos.

2.2.3 Interfaz de usuario

The image shows a login interface for 'Censo Uruguay'. At the top left is a logo with the text '2023 Censo URUGUAY'. Below the logo, the word 'Login' is displayed. There are two input fields: 'Nombre Usuario' and 'Contraseña'. Below these fields are two buttons: 'Ingresar' and 'Registrar'.

Validaciones

Los censistas y las credenciales de acceso deberán estar precargados en la aplicación o registrados en el momento, por lo que se deberá validar contra esos datos. En caso de error se debe informar al usuario en la propia interfaz. El ingreso del nombre de usuario podrá ser case insensitive, pero la contraseña deberá coincidir en mayúsculas y minúsculas. Todos los datos son obligatorios.

2.3 F03 – Ingresar datos de una persona Censada

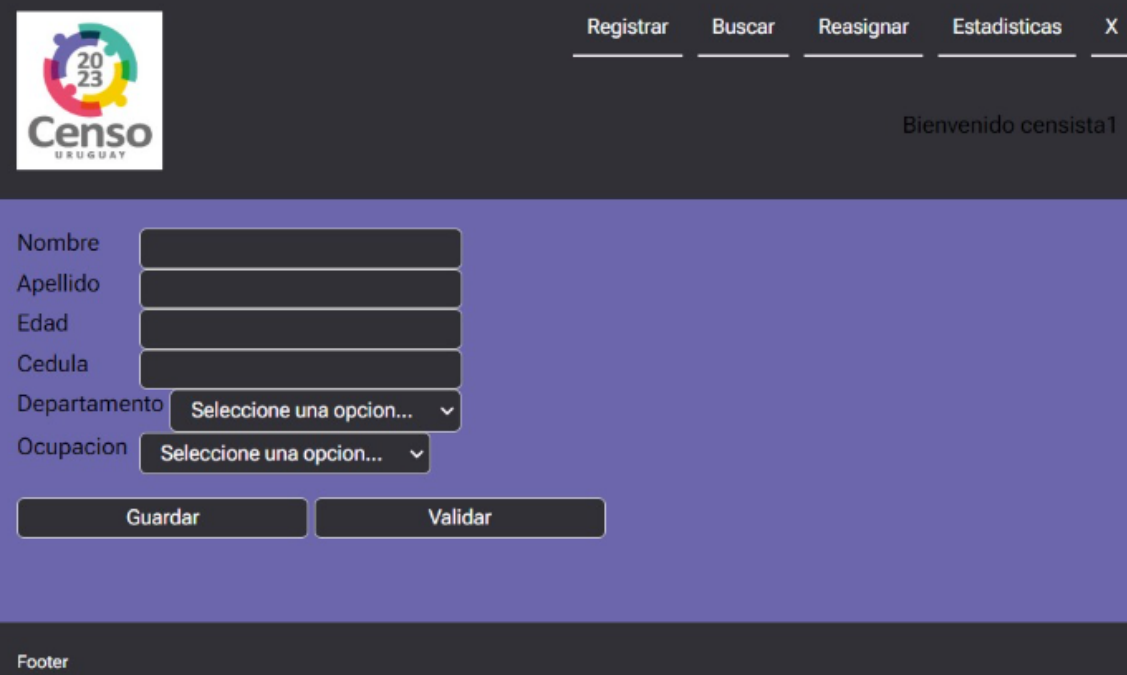
2.3.1 Acceso

Censista

2.3.2 Descripción

El censista podrá ingresar los datos para censar a una persona. Solicitándole el nombre, edad, cedula, departamento de residencia y ocupación.

2.3.3 Interfaz de Usuario



The screenshot displays the 'Censo Uruguay' web application interface. At the top left is the logo for the 2023 Census of Uruguay. To the right of the logo is a navigation bar with five links: 'Registrar', 'Buscar', 'Reasignar', 'Estadísticas', and 'X'. Below the navigation bar, the text 'Bienvenido censista1' is visible. The main form area has a light blue background and contains the following fields: 'Nombre' (text input), 'Apellido' (text input), 'Edad' (text input), 'Cedula' (text input), 'Departamento' (dropdown menu with 'Seleccione una opcion...' as the placeholder), and 'Ocupacion' (dropdown menu with 'Seleccione una opcion...' as the placeholder). Below these fields are two buttons: 'Guardar' and 'Validar'. At the bottom left of the interface is a 'Footer' label.

Validaciones

Todos los datos para ingresar son obligatorios, la persona solo será censada una vez por eso se validará que no esté ingresada dos veces.

2.4 F04 - Consultar censos pre completados

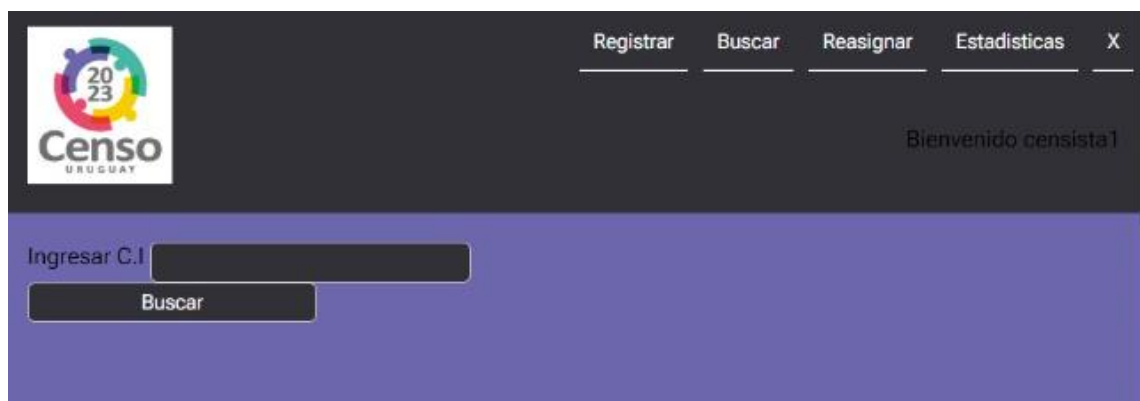
2.4.1 Acceso

Censista

2.4.2 Descripción

El censista puede llegar a una casa y validar que la persona ya pre completó sus datos para el censo. En ese caso el usuario buscará a la persona por su cédula, verificando si en efecto completó correctamente los datos. En caso de que esté todo correcto, el censista modificará los datos simplemente cambiando la propiedad censada.

2.4.3 Interfaz



The screenshot shows the web interface for the 'Censo Uruguay' system. At the top, there is a navigation bar with the following links: 'Registrar', 'Buscar', 'Reasignar', 'Estadísticas', and 'X'. The 'Buscar' link is highlighted. Below the navigation bar, the text 'Bienvenido censista1' is displayed. The main content area has a purple background and contains a form with the label 'Ingresar C.I.' followed by a text input field. Below the input field is a button labeled 'Buscar'.

Validaciones.

En esta opción funcionalidad no hay datos obligatorios. El usuario solo debe ingresar un dato correcto, sino dará aviso de error o de que ese usuario no existe.

2.5 F05 – Reasignar personas para validar

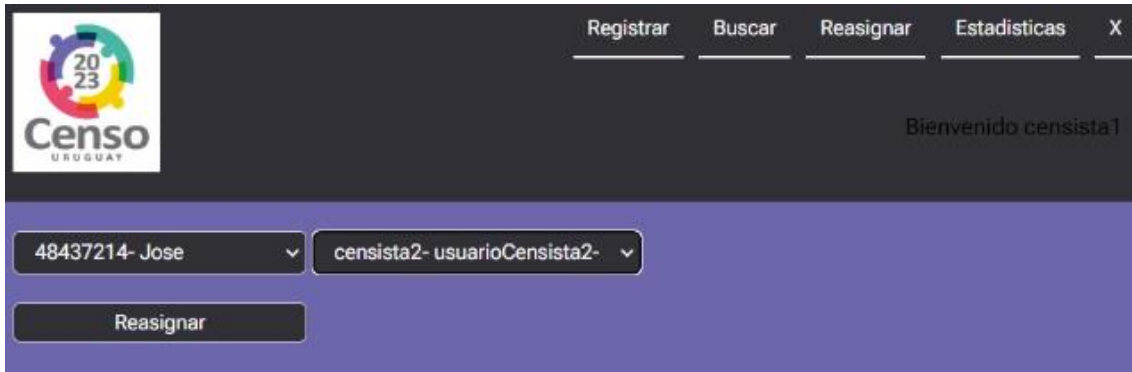
2.5.1 Acceso

Censista

2.5.2 descripción

El censista podrá seleccionar alguna persona que esté pendiente de validar datos y que le haya sido asignada y reasignarla a otro censista

2.5.3 Interfaz



The screenshot shows the 'Censo Uruguay' web application interface. At the top, there is a navigation bar with the following links: 'Registrar', 'Buscar', 'Reasignar', 'Estadísticas', and 'X'. The 'Reasignar' link is highlighted. Below the navigation bar, the user is greeted with 'Bienvenido censista1'. The main content area features two dropdown menus: the first is labeled '48437214- Jose' and the second is labeled 'censista2- usuarioCensista2-'. Below these dropdowns is a button labeled 'Reasignar'.

Validaciones

No hay datos obligatorios, solo se podrá seleccionar personas que estén pendientes de validar, si ya fue validada no se mostrara para asignar.

2.6 - F06 – Visualizar información estadística

2.6.1 – Acceso

Censista

2.6.2 descripción

En esta funcionalidad se podrá ver de forma sencilla distintos indicadores como, por ejemplo: total de personas censadas, cantidad de personas censadas por

departamentos, personas pendientes por validar y también tendrá la opción de ver por departamento las personas mayores y menores de edad censadas.

2.6.3 Interfaz

Validaciones

No hay datos obligatorios. A efectos de las estadísticas solamente se tomarán las personas cuyos censos estén validados.



2.7- F07 – Registro en la aplicación

2.7.1 Acceso

Invitado (persona)

2.7.2 Descripción

Registrarse en la aplicación completando los campos: Nombre y Apellido- Edad- Cédula- Departamento de residencia- Ocupación

2.7.3 Interfaz



The screenshot shows a web interface for the 'Censo Uruguay' (Uruguay Census). At the top left is the logo, which features a colorful circular graphic with the year '2023' and the text 'Censo URUGUAY'. Below the logo, on a light blue background, are several input fields for user registration: 'Nombre' (Name), 'Apellido' (Surname), 'Edad' (Age), 'Cedula' (ID Card), 'Departamento' (Department), and 'Ocupacion' (Occupation). The 'Departamento' and 'Ocupacion' fields are dropdown menus with the placeholder text 'Seleccione una opcion...'. Below these fields is a green 'Guardar' (Save) button. At the bottom of the form area, the word 'Footer' is visible.

Validaciones

Todos los datos son obligatorios a completar, y en caso de estar registrado mostrar un mensaje de error

2.8– F08 – Búsqueda de usuario

2.8.1 Acceso

Usuario: Invitado(persona)

2.8.2. descripción

Contará con la funcionalidad de buscar si completo sus datos. Para eso escribirá su cédula en un campo y se realizará la búsqueda. Si el usuario ya fue censado se le mostrará un mensaje de error. Si el usuario no fue censado, pero ya había

precargado su información previamente se le precargarán campos de texto y combos desplegables con la información que registró en su momento para poder editarla.

2.8.3 Interfaz



The screenshot displays the 'Censo Uruguay' web interface. At the top left is the logo for the 2023 Censo Uruguay. Below the logo, there is a button labeled 'Precargar datos para el Censo'. A horizontal line separates this from the 'Buscar Registro' section. In this section, there is a label 'Ingresar C.I.' followed by a text input field. Below the input field is a button labeled 'Buscar'. Another horizontal line is below the search section, followed by a button labeled 'Info estadísticas de censo'.

Validaciones:

Para eso escribirá su cédula en un campo y se realizará la búsqueda

2.9– F09 Asignación Censista

2.9.1 Acceso

Usuario: Invitado (persona)

2.9.2 descripción

Se le asignará un censista a cada usuario registrado, en el momento que se haga esta asignación, al usuario persona se le mostrará el nombre del censista que va a ir a su domicilio.

2.9.3 Interfaz

Usuario registrado con éxito!

A "usuario" se le asignó el censista "censista"

Validaciones:

Completar los datos y será asignada a un censista aleatoriamente

2.10- F10 - Eliminar sus datos para el censo

2.10.1 Acceso

Usuario: Invitado (persona)

2.10.2 descripción

El usuario deberá poder eliminar definitivamente los datos que había ingresado para el censo.

2.10.3 interfaz

The screenshot shows a web interface with a purple background. At the top, there is a header 'Buscar Registro'. Below it, there is a form with a label 'Ingresar C.I.' and a text input field containing the value '48437214'. To the right of the input field is a button labeled 'Buscar'. Below the form, there is a list of details for a record, each preceded by a bullet point: 'Nombre: Jose', 'Apellido: Rodriguez', 'Edad: 27', 'Departamento: Canelones', and 'Ocupacion: independiente'. At the bottom of the list is a button labeled 'Eliminar'. The footer of the page is labeled 'Footer'.

Validaciones:

Solo se podrán eliminar los datos en caso de que no se haya validado ese censo.

2.11– F11 - Lista de censados

2.11.1 Acceso

Usuario: Invitado (persona)

2.11.2 descripción

Se deberá mostrar un listado en formato tabla con la información proporcionada.

2.11.3 interfaz

Info estadísticas de censo				
Departamento	Estudian	No trabajan	Dependientes o Independientes	Porcentaje del total de censados
Montevideo	1	0	1	6.90 %
Canelones	0	1	1	6.90 %
Florida	0	0	2	6.90 %
Rocha	1	0	0	3.45 %
Maldonado	0	1	0	3.45 %
Tacuarembó	0	1	1	6.90 %
Paysandú	0	0	2	6.90 %
Flores	1	0	1	6.90 %
Durazno	1	0	0	3.45 %
Treinta y tres	0	1	1	6.90 %
Salto	0	1	0	3.45 %
Artigas	1	0	1	6.90 %
Rio Negro	0	0	1	3.45 %
LaValleja	1	0	0	3.45 %
Cerro Largo	0	1	0	3.45 %
Rivera	0	0	1	3.45 %
San Jose	1	0	1	6.90 %
Soriano	0	1	1	6.90 %
Colonia	0	0	1	3.45 %

Validaciones

Debe existir información cargada para listarla

3. Informe de Testing

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A REGISTRARSE EN LA APLICACIÓN USUARIO CENSITA

F01: Registrarse en la aplicación.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F01 – T01	Completar datos para registrarse	1.Selecciona en el menú, la opción “registro” 2.Completa todos los campos correctamente 3.Confirma en “confirmar registro”	Nombre: censista1 Nombre usuario: usuarioCensista1 Contraseña: Em12345	Registro exitoso	correcto	P
F01 – T02	Algunos de los campos vacíos	1.Algunos de los campos no se ingresa dato	Nombre: “vacio” Nombre Usuario: usuarioCensista1 Contraseña: Em12345	Mensaje error “Todos los campos deben ser completados”	correcto	P
F01 – T03	Contraseña menor a 5 caracteres sin mayúscula y sin números	1. En el campo contraseña ingresa una no valida	Nombre: censita1 Nombre usuario: usuarioCensista1 Contraseña: “cens”	Mensaje error “contraseña incorrecta”	correcto	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIADAS A INGRESAR EN LA APLICACIÓN USUARIO CENSITA.

F02 - Ingresar en la aplicación

ID	Escenario de prueba	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F = falla, P = pasa)
F01 – T01	Ingreso correcto	1.Ingresa nombre usuario 2.Ingresa contraseña	1.Usuario: usuarioCensista1 2.Contraseña: Em12345	Login exitoso	Login exitoso	P
F01 – T02	Usuario no encontrado	1.Ingresa nombre 2.Ingresa contraseña	1. Usuario: usuarioCen1234 2. Contraseña: Pep1234	Usuario incorrecto,	No permite logearse, y mensaje de error “usuario no existe”	p
F01 – T03	Contraseña incorrecta	1. Ingresa nombre 2. Ingresa contraseña	1.Usuario: usuarioCensista1 2.Contraseña: Em123	Contraseña incorrecta	Mensaje error: “contraseña incorrecta”	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS INGRESAR DATOS DE UNA PERSONA CENSADA

F03 - Ingresar datos de una persona Censada.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F03 – TO1	Ingresar datos para censar una persona	1. Seleccionar en el menú “registrar” 2. Completar todos los campos 3. Confirmar en “guardar”	1. Nombre: Agustin 2. Apellido: Torres 3. Edad; 27 4. Cedula: 50653009 5. Departamento: Canelones 6. Ocupacion: Estudiante	Censo exitoso	Ok	P
F03 – TO2	Uno de los campos no se completa	1. Seleccionar en el menú “registrar” 2. Uno de los campos no se completa 3. Confirmar en “guardar”	1. Nombre: Agustin 2. Apellido: “vacío” 3. Edad; 27 4. Cedula: 50653009 5. Departamento: Canelones 6. Ocupacion: Estudiante	Mensaje error: “debe completar todos los campos”	ok	P
F03 – TO3	Persona ya fue censada	1. Seleccionar en el menú “registrar” 2. Completar todos los campos 3. Confirmar en “guardar”	1. Nombre: Ema 2. Apellido: Rodr 3. Edad; 27 4. Cedula: 48433413 5. Departamento: Paysandu 6. Ocupacion: Estudiante	Mensaje de error “persona ya censada”	ok	P
F03 – TO4	Cedula no valida	1. Seleccionar en el menú “registrar” 2. Completar todos los	1. Nombre: Agustin 2. Apellido: Torres	Mensaje de error “cedula no valida”	Ok	P

		campos – campo cedula no valida 3. Confirmar en “guardar”	3.Edad: 27 4.Cedula: 5065309 5.Departamento: Canelones 6.Ocupacion: Estudiante			
--	--	--	---	--	--	--

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS CONSULTAR CENSOS PRE COMPLETADOS

F04 – Consultar censos pre completados

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F04 – T01	Consultar censo pre completados para validar	1.En el campo para ingresar CI, se ingresa la CI a consultar 2.Click en “buscar” 3. Se muestran datos	1. Cedula: 50653009	Se muestran los datos pre completados	Ok	P
F04 – T02	Se ingresa una cedula de un usuario que no pre completo o mal ingresada	1.En el campo para ingresar CI, se ingresa la CI a consultar 2.Clic en “buscar” 3. Se muestra mensaje de error	1. Cedula: 50653003	Se muestra mensaje de error	Ok	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS REASIGNAR PERSONA PARA VALIDAR

F05 – Reasignar persona para validar

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F05 – T01	Reasignar a una persona que completo sus datos a otro censista	1.De combo desplegable se seleccionará a una persona 2.Se selecciona del combo desplegable donde están los censistas 3. Se da click en el botón “reasignar”	1.Combo desplegable personas: “48437214 – Jose” 2. Combo desplegable censista: “censista2 – usuarioCensista2”	Reasignación exitosa	Ok	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A VISUALIZAR INFORMACION ESTADISTICA

F06 – Visualizar información estadística.

Precondiciones: - No hay datos obligatorios. A efectos de las estadísticas solamente se tomarán las personas cuyos censos estén validados.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F06 – T01	Mostrar el total de las personas censadas	1.Clic en el botón “estadísticas” 2. Click en el botón “mostrar” en el campo “total personas censadas”	Todos los censos validos	Ver info.	Correcto	P
F06 – T02	Mostrar total personas	1.Clic en el botón “estadísticas”	Todos los censos validados	Ver info.	Correcto	P

	censadas por depto.	2. Click en el botón "mostrar" en el campo "total personas censadas por depto."				
F06 – T03	Mostrar % de personas pendientes	1.Clic en el botón "estadísticas" 2. Click en el botón "mostrar" en el campo "mostrar % de personas pendientes"	Todos los censos no validados	Ver info	Correcto	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A REGISTRO EN LA APLICACIÓN USUARIO INVITADO

F07 – Registro en la aplicación usuario invitado

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F07 – T01	Registrar persona de forma correcta	1.Completar todos los campos 2. seleccionar "Guardar"	1.Nombre: Agustin 2.Apellido: Torres 3.Edad: 27 4.Cedula: 5065309 5.Departamento: Canelones 6.Ocupacion: Estudiante	Registro exitoso	Correcto	P
F07 – T02	Cedula ingresada no valida - formato no correcto	1.Completar los campos 2.El campo cedula se ingresa con una cedula no valida	1.Nombre: Agustin 2.Apellido: Torres 3.Edad: 27 4.Cedula: 50309 5.Departamento: Canelones 6.Ocupacion:	Mensaje de error "cedula no valida"	Correcto	P

			Estudiante			
F07 – T03	Edad ingresada mayor a 130 años	1.Completa los campos 2.El campo edad mayor a 130 años	1.Nombre: Agustin 2.Apellido: Torres 3.Edad: 12 4.Cedula: 50309 5.Departamento: Canelones 6.Ocupacion: Estudiante	Mensaje de error “Edad no validez”	Correcto	P
F07-T04	El usuario invitado ya se había registrado	1.Completa todos los campos	1.Nombre: Agustin 2.Apellido: Torres 3.Edad: 27 4.Cedula: 50309 5.Departamento: Canelones 6.Ocupacion: Estudiante	Mensaje de error “usuario ya precargo sus datos”	Correcto	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIADAS A LA BÚSQUEDA DE USUARIO

F08 – Búsqueda de usuario

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F08 – T01	Buscar si pre complete los datos mediante el ingreso de mi CI	1.Ingreso CI en el campo buscar 2.Click en “buscar” 3.Guardar	Ci: “48437214”	Se muestra los datos que pre completo y puede completarlos y guardar	Correcto	P

F08 T02	–	Usuario ya fue validado por un censista	1.Ingreso CI en el campo buscar 2.Click en “buscar”	Ci: “48437244”	Se muestra el mensaje “persona ya censada”	Correcto	P
F08 T03	–	Usuario no fue censado, pero si pre-completo los datos	1.Ingreso CI en el campo buscar 2.Click en “buscar”	CI: “48438214”	Se muestra los campos para pre completar los datos y los guarda	Correcto	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A LA ASIGNACION AUTOMATICA DE CENSISTA

F09 – Asignación automática de censista

Precondición: - se le asigna censista de forma automática siempre que el usuario complete y guarde sus datos.

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F09 – T01	Al momento del usuario invitado completa su registro se le asignan automáticamente un censista	1.completar todos los datos correctamente 2. clic en guardar	1.Nombre: Paula 2.Apellido: Torres 3.Edad: 27 4.Cedula: 5065303-9 5.Departamento: Canelones 6.Ocupacion: Estudiante	Mensaje de éxito, “al usuario se le asigno el siguiente censista”	Correcto	P

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A ELIMINAR SUS DATOS PARA EL CENSO

F10 – Eliminar sus datos para el censo

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado
F10 – T01	Eliminar los datos pre completados que no validaron.	1. Ingresar en el campo buscada la CI 2. Click en buscar. 3. Por último “Eliminar”	Ci: “48437244”	Se muestra la información del usuario y le permite eliminar
F10 – T02	El usuario ingreso su CI, pero ya fue validada por un censista	1.ingresa si CI 2.Busca	Ci: “48437214”	Se muestra el mensaje “ci ya validada, no se puede eliminar”
F10 – T03	Usuario ingresa una CI no valida	1.ingresa CI 2.Clic en buscar	Ci: “50653009”	Se muestra el mensaje “Ci no valida, volver a ingresar”

CASOS DE PRUEBA PARA FUNCIONALIDADES ASOCIDAS A LISTA DE CENSADOS

F11 – Listado de censados

ID	Escenario de test	Pasos	Datos utilizados	Resultado esperado	Resultado obtenido	Estado (F=Falla, P=pasa)
F11 – T01	Se mostrará un listado de la info estadística por departamento al momento de hacer click en “mostrar info estadística”	1.Una vez logueado como usuario invitado hacer click en “mostrar info estadística” 2.Se despliega la tabla	1.Ingresar con cualquier dato precargado	Mostrar tabla con la información	Correcto	P

4. Código de desarrollo

4.1 Lista de Pre-Cargas de datos

Precarga de datos Censista		
Nombre	Nombre usuario	contraseña
censista-1	usuarioCensista1	Em12345
censista-2	usuarioCensista2	Em12345
censista-3	usuarioCensista3	Em12345

Precarga de datos censados						
Nombre	Apellido	Edad	CI	Departamento	ocupación	Estado
Jose	Rodriguez	27	48437214	Canelones	independiente	false
Pepe	Torres	14	48437244	San Jose	independiente	true
Juan	Perdomo	21	48438214	Artigas	estudiante	false
Julio	Almeida	29	45437214	Tacuarembó	no trabaja	false
Javier	Alfonso	33	45537264	Paysandu	dependiente	true
Emanuel	Rodriguez	12	48433415	Flores	independiente	true
Florencia	Casas	13	64756987	Florida	dependiente	true
Antonia	Ramirez	14	53678904	Durazno	estudiante	true
Alfonsa	Paredes	16	32358763	Soriano	no trabaja	true
Rosita	Cardozo	8	47652341	Colonia	independiente	true
Pedro	Alfonso	63	55361343	Treinta y tres	independiente	false
Julian	Alvarez	46	39457653	Montevideo	estudiante	false
Luis	Suarez	36	43651341	Canelones	no trabaja	true
Facundo	Torres	63	82568741	Florida	dependiente	true
Antonio	Pacheco	19	24658563	Montevideo	independiente	false
Alberto	Sellanes	63	43675462	Rocha	estudiante	true
Obdulio	Varela	89	25679862	Maldonado	no trabaja	false
Nestor	Goncalves	24	34569876	Tacuarembó	dependiente	false
kevin	Dawson	33	25438769	Paysandu	independiente	false
Valentin	Rodriguez	13	65423456	Flores	estudiante	true
Dario	Rodriguez	8	38519238	Treinta y tres	no trabaja	false
Sebastian	Cristoforo	12	25673459	Salto	no trabaja	true
Fabian	Estoyanoff	59	67452463	Artigas	dependiente	true
Cristiano	Ronaldo	13	36457653	Rio Negro	independiente	false
Ronaldinho	Gaucha	23	24957432	LaValleja	estudiante	false
Lionel	Messi	19	67859664	Cerro Largo	no trabaja	false
Neymar	J.R	66	45679863	Soriano	independiente	true
Ronaldo	Nazario	91	53425436	Rivera	dependiente	false
Darwin	Nuñez	12	29876543	San Jose	estudiante	true
Luis	Nuñez	12	29876665	San Jose	estudiante	true

4.2 Código HTML:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <title>Obligatorio</title>
    <script src="js/Clases.js"></script>
    <script src="js/Sistema.js"></script>
    <script src="js/Codigo.js"></script>

    <link rel="stylesheet" href="css/estilos.css" />
  </head>

  <body>
    <div id="contenedor">
      <div id="cabeza1" class="clearfix">
        

        <ul id="navPrincipal">
          <li id="btnMostrarRegistrar">
            <a>Registrar</a>
          </li>

          <li id="btnBuscar">
            <a>Buscar</a>
          </li>

          <li id="btnReasignarPersona">
            <a>Reasignar</a>
          </li>

          <li id="btnEstadisticas">
            <a>Estadisticas</a>
          </li>

          <li id="btnSalir">
            <a>X</a>
          </li>
        </ul>
      </div>
    </div>
  </body>
</html>
```

```

        </li>
    </ul>
    <div id="aux">
        <p id="nombreUsuarioLogeado"></p>
    </div>
</div>
<!-- FIN CABEZAL -->

<div id="contenido">
    <!-- Seccion Perfiles-->
    <div id="seccionPerfiles">
        <h3>Seleccionar Perfil</h3>

        <select id="slcPerfiles">
            <option value="censista">Censista</option>
            <option value="persona">Invitado (Persona)</option>
        </select>
        <br /><br />

        <input type="button" id="btnIngresarCensista" value="Ingresar"
/>
    </div>
    <!-- Seccion Login-->
    <div id="seccionLogin">
        <h3>Login</h3>
        <label for="loginUsuario">Nombre Usuario</label>
        <input type="text" id="loginUsuario" /><br />

        <label for="loginClave">Contraseña</label>
        <input type="password" id="loginClave" /><br />
        <br />

        <input type="button" id="btnLogin" value="Ingresar" />

        <input type="button" id="btnRegistrar" value="Registrar" />
        <p id="mensajeLogin"></p>
    </div>
    <!-- Seccion Registrar Usuario-->
    <div id="seccionRegistrarUsuario">
        <h3>Registro</h3>
        <label for="nombreCen">Nombre</label>
        <input type="text" id="nombreCen" /><br />

        <label for="nombreUsuario">Nombre Usuario</label>
        <input type="text" id="nombreUsuario" /><br />

        <label for="clave">Contraseña</label>

```

```

<input type="password" id="clave" /><br />
<br />
<input
  type="button"
  id="confirmarRegistro"
  value="Confirmar Registro"
/>
<p id="mensajeRegistro"></p>
</div>
<!-- Seccion Reasignar Censo-->
<!-- Hacer una class seccion para ocultar CONTENIDO -->
<div id="seccionRegistrarCenso">
  <label for="nombreRegistroCenso">Nombre</label>
  <input type="text" id="nombreRegistroCenso" /><br />

  <label for="apellidoRegistroCenso">Apellido</label>
  <input type="text" id="apellidoRegistroCenso" /><br />

  <label for="edadRegistroCenso">Edad</label>
  <input type="text" id="edadRegistroCenso" /> <br />

  <label for="cedulaRegistroCenso">Cedula</label>
  <input type="text" id="cedulaRegistroCenso" /><br />

  <label for="slcDepartamento">Departamento</label>
  <select id="slcDepartamento"></select>
  <br />

  <label for="slcOcupacion">Ocupacion</label>
  <select id="slcOcupacion"></select>
  <br />

  <p id="mensajeRegistroCenso"></p>

  <input type="button" value="Guardar" id="btnRegistrarCenso" />
  <input type="button" value="Validar" id="btnValidarCenso" />
  <input type="button" value="Guardar" id="btnGuardarInvitado" />

  <!-- -->
</div>
<!-- Seccion Buscar-->
<div id="seccionBuscador">
  <label for="buscarCedula">Ingresar C.I</label>
  <input type="text" id="buscarCedula" /><br />
  <input type="button" id="btnBuscarCi" value="Buscar" />
  <p id="pResultadoBusqueda"></p>
  <input type="button" value="Validar" id="btnValidarCensos" />

```

```

        <p id="pResultadoValidadoEnBusqueda"></p>
    </div>
    <!-- Seccion Reasignar -->
    <div id="seccionReasignar">
        <select id="slcReasignar"></select>
        <select id="slcCensistas"></select>
        ><br />
        <br />

        <input type="button" id="btnReasignarCensado" value="Reasignar"
/>

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

<!-- Seccion Info Estadistica -->
<div id="seccionInfoEstadistitca">
    <label for="totalCensadas">Total personas Censadas: </label>
    <input type="button" id="idTotalCensados" value="Mostrar" />
    <p id="pTotalCensados"></p>
    <br />
    <label for="totalCensadasporDepto"
        >Total personas Censadas por departamento:
    </label>
    <input type="button" id="idTotalporDepto" value="Mostrar" /> <br
/>

    <table id="tabla-conteo" border="1"></table>
    <p id="pTotalCensadosDepto"></p>
    <br />
    <label for="totalPendientesporValidar"
        >Total personas pendintes:
    </label>
    <input type="button" id="idPendites" value="Mostrar" />
    <p id="pTotalpendientes"></p>
    <br />
    <!--Select seleccionar depto-->
    <select id="slcSeleccionarDepto2"></select>
    ><br />
    <input
        type="button"
        value="Seleccionar departamentos"
        id="btnVerDeptos"
    />
    <input
        type="button"
        value="Mostrar Estadistica"
        id="btnMostrarEstadisticas"

```

```

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

<!--Seccion Invitados-->
<div id="seccionInvitados">
    <br />
    <input
        type="button"
        value="Precargar datos para el Censo"
        id="btnRegistrarInvi"
    />
    <br />
    <br />
    <hr />
    <h3 id="h3">Buscar Registro</h3>
    <label for="buscarCedulainv">Ingresar C.I</label>
    <input type="text" id="buscarCedulainv" /> <br />
    <input type="button" id="btnBuscarCiinv" value="Buscar" />
    <input type="button" value="Guardar" id="btnGuardarCensosinvi" />
<br />
    <br />
    <p id="pResultadoBusquedainv"></p>
    <hr />
    <input
        type="button"
        value="Info estadísticas de censo"
        id="btnMostrarEstadistica"
    />
    <table id="tablaInvitado"></table>
</div>

<!-- FIN CONTENIDO -->
<div id="pie">
    <a href="https://www.ort.edu.uy">Footer</a>
</div>
<!-- FIN PIE -->
</div>
<!-- FIN CONTENEDOR -->
</body>
</html>

```

4.4 Código Sistema.js

```
class Sistema {
  constructor() {
    this.censistas = [
      new Censista("censista-1", "usuarioCensista1", "Em12345", 1),
      new Censista("censista-2", "usuarioCensista2", "Em12345", 2),
      new Censista("censista-3", "usuarioCensista3", "Em12345", 3)
    ];

    this.censados = [
      new Censado("Jose", "Rodriguez", 27, 48437214, "Canelones",
        "independiente", 1, 1, false),
      new Censado("Pepe", "Torres", 14, 48437244, "San Jose",
        "independiente", 2, 2, true),
      new Censado("Juan", "Perdomo", 21, 48438214, "Artigas",
        "estudiante", 3, 3, false),
      new Censado("Julio", "Almeida", 29, 45437214, "Tacuarembó",
        "no trabaja", 1, 4, false),
      new Censado("Javier", "Alfonso", 33, 45537264, "Paysandu",
        "dependiente", 2, 5, true),
      new Censado("Emanuel", "Rodriguez", 12, 48433415, "Flores",
        "independiente", 3, 6, true),
      new Censado("Florencia", "Casas", 13, 64756987, "Florida",
        "dependiente", 1, 7, true),
      new Censado("Antonia", "Ramirez", 14, 53678904, "Durazno",
        "estudiante", 2, 8, true),
      new Censado("Alfonsa", "Paredes", 16, 32358763, "Soriano",
        "no trabaja", 3, 9, true),
      new Censado("Rosita", "Cardozo", 8, 47652341, "Colonia",
        "independiente", 1, 10, true),
      new Censado("Pedro", "Alfonso", 63, 55361343, "Treinta y
        tres", "independiente", 2, 11, false),
      new Censado("Julian", "Alvarez", 46, 39457653, "Montevideo",
        "estudiante", 3, 12, false),
      new Censado("Luis", "Suarez", 36, 43651341, "Canelones", "no
        trabaja", 1, 13, true),
      new Censado("Facundo", "Torres", 63, 82568741, "Florida",
        "dependiente", 2, 14, true),
      new Censado("Antonio", "Pacheco", 19, 24658563, "Montevideo",
        "independiente", 3, 15, false),
      new Censado("Alberto", "Sellanes", 63, 43675462, "Rocha",
        "estudiante", 1, 16, true),
      new Censado("Obdulio", "Varela", 89, 25679862, "Maldonado",
        "no trabaja", 2, 17, false),
```



```

        new Censado("Nestor ", "Goncalves", 24, 34569876,
"Tacuarembó", "dependiente", 3, 18, false),
        new Censado("kevin", "Dawson", 33, 25438769, "Paysandu",
"independiente", 1, 19, false),
        new Censado("Valentin", "Rodriguez", 13, 65423456, "Flores",
"estudiante", 2, 20, true),
        new Censado("Dario", "Rodriguez", 8, 38519238, "Treinta y
tres", "no trabaja", 3, 21, false),
        new Censado("Sebastian", "Cristoforo", 12, 25673459, "Salto",
"no trabaja", 1, 22, true),
        new Censado("Fabian", "Estoyanoff", 59, 67452463, "Artigas",
"dependiente", 2, 23, true),
        new Censado("Cristiano", "Ronaldo", 13, 36457653, "Rio
Negro", "independiente", 3, 24, false),
        new Censado("Ronaldo", "Gaucho", 23, 24957432,
"LaValleja", "estudiante", 1, 26, false),
        new Censado("Lionel", "Messi", 19, 67859664, "Cerro Largo",
"no trabaja", 2, 27, false),
        new Censado("Neymar", "J.R", 66, 45679863, "Soriano",
"independiente", 3, 28, true),
        new Censado("Ronaldo", "Nazario", 91, 53425436, "Rivera",
"dependiente", 1, 29, false),
        new Censado("Darwin", "Nuñez", 12, 29876543, "San Jose",
"estudiante", 2, 30, true),
        new Censado("Luis", "Nuñez", 12, 29876665, "San Jose",
"estudiante", 2, 31, true)
    ];

    this.ocupaciones = [
        new Ocupacion(1, "dependiente"),
        new Ocupacion(2, "independiente"),
        new Ocupacion(3, "estudiante"),
        new Ocupacion(4, "no trabaja")
    ];

    this.departamentos = [
        new Departamento(1, "Montevideo"),
        new Departamento(2, "Canelones"),
        new Departamento(3, "Florida"),
        new Departamento(5, "Rocha"),
        new Departamento(6, "Maldonado"),
        new Departamento(7, "Tacuarembó"),
        new Departamento(8, "Paysandu"),
        new Departamento(9, "Flores"),
        new Departamento(10, "Durazno"),
        new Departamento(11, "Treinta y tres"),
        new Departamento(12, "Salto"),
    ];

```

```

        new Departamento(13, "Artigas"),
        new Departamento(14, "Rio Negro"),
        new Departamento(15, "LaValleja"),
        new Departamento(16, "Cerro Largo"),
        new Departamento(17, "Rivera"),
        new Departamento(18, "San Jose"),
        new Departamento(19, "Soriano"),
        new Departamento(20, "Colonia")
    ];

}

contarPersonasPorDepartamentoInvi() {
    let conteo = {};
    for (let i = 0; i < this.departamentos.length; i++) {
        let depto = this.departamentos[i].nombre;
        conteo[depto] = {
            estudiantes: 0,
            noTrabajan: 0,
            dependienteOIndependiente: 0,
            totalCensados: 0,
            porcentaje: 0,
        };
    }

    for (let j = 0; j < this.censados.length; j++) {
        const censo = this.censados[j];
        let departamento = censo.depto;
        let ocupacion = censo.ocupacion;

        if (ocupacion === 'estudiante') {
            conteo[departamento].estudiantes++;
        } else if (ocupacion === 'no trabaja') {
            conteo[departamento].noTrabajan++;
            conteo[departamento].dependiente++;
        } else if (ocupacion === 'dependiente' || ocupacion ===
'indeependiente') {
            conteo[departamento].dependienteOIndependiente++;
        }

        conteo[departamento].totalCensados++;
    }

    let totalCensados = this.censados.length;
    for (let depto in conteo) {
        conteo[depto].porcentaje = ((conteo[depto].totalCensados /
totalCensados) * 100).toFixed(2) + " %"
    }
}

```

```

    }

    return conteo;
}

agregarCensista(censista) {
    this.censistas.push(censista)
}

agregarCensados(censista) {
    this.censados.push(censista)
}

buscarElemento(arrElementos, propiedad, busqueda) {
    let existe = false;
    for (let i = 0; i < arrElementos.length; i++) {
        const unElemento = arrElementos[i];
        if (unElemento[propiedad] === busqueda) {
            existe = true;
            break;
        }
    }
    return existe;
}

obtenerObjeto(arrElementos, propiedad, busqueda) {
    let objeto = null;
    for (let i = 0; i < arrElementos.length; i++) {
        const unElemento = arrElementos[i];
        if (unElemento[propiedad] == busqueda) {
            objeto = unElemento;
            break
        }
    }
    return objeto;
}

validarCensistaRepetido(censista) {
    let existe = false;
    for (let i = 0; i < this.censistas.length; i++) {
        if (censistas[i] === censista) {
            existe = true;
            break
        }
    }
    return existe;
}

```

```

    }
    //---Valida el registro de usuario censista
    validarRegistroOk(nombre, nombreUsuario, clave) {
        let campoValido = false;
        if (nombre != "" && nombreUsuario != "" && clave != "") {
            campoValido = true;
        }
        return campoValido;
    }
    validarContraseña(contraseña) {
        if (contraseña.length < 5) {
            return false;
        }
        let tieneMayuscula = false;
        let tieneMinuscula = false;
        let tieneNumero = false;

        //validamos para que pueda ingresar mayus, min
        const caracteresMayusculas = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        const caracteresMinusculas = "abcdefghijklmnopqrstuvwxyz";
        const caracteresNumeros = "0123456789";

        for (let i = 0; i < contraseña.length; i++) {
            const caracter = contraseña[i];

            if (caracteresMayusculas.indexOf(caracter) !== -1) {
                tieneMayuscula = true;
            }
            if (caracteresMinusculas.indexOf(caracter) !== -1) {
                tieneMinuscula = true;
            }
            if (caracteresNumeros.indexOf(caracter) !== -1) {
                tieneNumero = true;
            }
            // Verificar si la contraseña es valida
            if (tieneMayuscula && tieneMinuscula && tieneNumero) {
                return true;
            }
        }
        return false;
    }

    verificarLogin(nombreUsuario, clave) {
        let resultado = false;
        let unUsuario = this.obtenerObjeto(this.censistas,
"nombreUsuario", nombreUsuario);
        if (unUsuario !== null) {

```

```

        if (unUsuario.contrasena === clave) {
            resultado = true;
        }
    }
    return resultado;
}

//Eliminar Censo completado (solo el invitado elimina si no fue
validado.)
eliminarCenso(pos) {
    this.censados.splice(pos, 1);
}

eliminarCaracter(texto, letra) {
    let textoSustituido = "";
    for (let i = 0; i < texto.length; i++) {
        if (texto.charAt(i) !== letra) {
            textoSustituido += texto.charAt(i);
        }
    }
    return textoSustituido;
}

//funcion estadisitca, cuenta la edad por departamento.
contarEdadPorDepartamento(censados, departamentos,
departamentoSeleccionado) {
    let conteoEdadMayores = 0;
    let conteoEdadMenores = 0;

    for (let i = 0; i < censados.length; i++) {
        let censo = censados[i];
        let departamento = censo.depto;

        if (departamento === departamentoSeleccionado) {
            if (censo.edad >= 18 && censo.estadoValidado !== false) {
                conteoEdadMayores++;
            } else if (censo.edad < 18) {
                conteoEdadMenores++;
            }
        }
    }

    return {
        cantidadMayores: conteoEdadMayores,
        cantidadMenores: conteoEdadMenores
    };
}

```

```

    contarPersonasPorDepartamento(censados, departamentos) {
        let conteo = {};
        // Inicializar el conteo en cero para cada departamento
        for (let i = 0; i < departamentos.length; i++) {
            let depto = departamentos[i].nombre;
            conteo[depto] = 0;
        }
        // Recorrer el array de censados y contar las personas por
departamento
        for (let j = 0; j < censados.length; j++) {
            let censo = censados[j];
            let departamento = censo.depto;

            if (censo.estadoValidado !== false)
                conteo[departamento]++;
        }
        return conteo;
    }

    validation_digit(ci) {
        var a = 0;
        var i = 0;
        if (ci.length <= 6) {
            for (i = ci.length; i < 7; i++) {
                ci = '0' + ci;
            }
        }
        for (i = 0; i < 7; i++) {
            a += (parseInt("2987634"[i]) * parseInt(ci[i])) % 10;
        }
        if (a % 10 === 0) {
            return 0;
        } else {
            return 10 - a % 10;
        }
    }

    clean_ci(ci) {
        return ci.replace(/\D/g, '');
    }

    validarCedula(ci) {
        ci = this.clean_ci(ci);
        var dig = ci[ci.length - 1];
        ci = ci.replace(/[0-9]$/, '');
        return (dig == this.validation_digit(ci));
    }
}

```

```

    aleatorioID(array) {
        let indiceAleatorio = Math.floor(Math.random() * array.length);
        return array[indiceAleatorio].id;
    }
}

```

4.5 Codigo.js

```

window.addEventListener("load", inicio);

function inicio() {
    //-----Esconde secciones-----
    document.querySelector("#seccionLogin").style.display = "none";
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none";
    document.querySelector("#seccionPerfiles").style.display = "block";
    document.querySelector("#navPrincipal").style.display = "none";
    document.querySelector("#seccionRegistrarCenso").style.display =
"none";
    document.querySelector("#seccionBuscador").style.display = "none";
    document.querySelector("#seccionReasignar").style.display = "none";
    document.querySelector("#seccionInfoEstadistica").style.display =
"none";
    document.querySelector("#seccionInvitados").style.display = "none";

    //-----Cargar combo desplegables-----
    cargarSlcDpto();
    cargarSlcOcup();

    //-----Funciones ocultar y mostrar secciones, y eventos de click---
    -----

    document.querySelector("#btnMostrarRegistrar").addEventListener("click",
CargarRegistroCenso);
    document.querySelector("#btnRegistrar").addEventListener("click",
mostrarRegistro);
    document.querySelector("#btnBuscar").addEventListener("click",
cargarBuscador);

    document.querySelector("#btnReasignarPersona").addEventListener("click",
mostrarSeccionReasignarCenso);
    document.querySelector("#btnEstadisticas").addEventListener("click",
mostrarSeccionEstadisticas);

```

```

        document.querySelector("#btnRegistrarInvi").addEventListener("click",
seccionRegistrarInvi);

document.querySelector("#btnMostrarEstadisticas").addEventListener("click",
", mostrarEstadisticasPorDepartamento);

document.querySelector("#btnMostrarEstadistica").addEventListener("click",
, tablaEstadisticaInvitado);

//-----Ecentos de los botones para las distintas funcionalidades-----
-----

document.querySelector("#btnIngresarCensista").addEventListener("click",
loginUsuario);
    document.querySelector("#btnLogin").addEventListener("click",
hacerLogin);
    document.querySelector("#btnSalir").addEventListener("click",
logout);

document.querySelector("#confirmarRegistro").addEventListener("click",
registrarCensista);

document.querySelector("#btnRegistrarCenso").addEventListener("click",
guardarCenso);
    document.querySelector("#btnValidarCenso").addEventListener("click",
validarCenso);
    document.querySelector("#btnValidarCensos").addEventListener("click",
validarEnBusquedaCensista);
    document.querySelector("#btnBuscarCi").addEventListener("click",
buscarCensado);
    document.querySelector("#btnBuscarCiinv").addEventListener("click",
buscarCiInvitado);

document.querySelector("#btnReasignarCensado").addEventListener("click",
reasignarCenso);
    document.querySelector("#idTotalCensados").addEventListener("click",
totalPersonasCensadas);
    document.querySelector("#idTotalporDepto").addEventListener("click",
censadosPorDepto);
    document.querySelector("#idPendites").addEventListener("click",
personasPendientes);
    document.querySelector("#btnVerDeptos").addEventListener("click",
seleccionarDeptosTotal);

document.querySelector("#btnGuardarInvitado").addEventListener("click",
guardarRegistroInvitados);
}

```



```

let sistema = new Sistema;

//-----Registro-----
let usuarioLogueado = null;
// Aca se muestra la pantalla cuando lueguea el usuario
function mostrarUsuarioLogueado() {
    document.querySelector("#seccionPerfiles").style.display = "none"
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none"
    document.querySelector("#seccionLogin").style.display = "none"
    document.querySelector("#nombreUsuarioLogeado").style.display =
"block"
    document.querySelector("#seccionBuscador").style.display = "none"
    document.querySelector("#seccionReasignar").style.display = "none"
    document.querySelector("#seccionInfoEstadisitca").style.display =
"none"
    document.querySelector("#nombreUsuarioLogeado").innerHTML =
"Bienvenido " + usuarioLogueado.nombre;
}

//-----Aca seleccionamos el tipo de perfil-----
function loginUsuario() {
    let perfil = document.querySelector("#slcPerfiles").value;
    if (perfil === "censista") {
        document.querySelector("#seccionLogin").style.display = "block";
        document.querySelector("#seccionPerfiles").style.display =
"none";
        document.querySelector("#seccionRegistrarUsuario").style.display
= "none";
        document.querySelector("#seccionReasignar").style.display =
"none";
        document.querySelector("#seccionInfoEstadisitca").style.display =
"none";
    }
    if (perfil === "persona") {
        document.querySelector("#seccionPerfiles").style.display =
"none";
        document.querySelector("#seccionInvitados").style.display =
"block";
        document.querySelector("#btnGuardarCensosinvi").style.display =
"none";
    }
}

//-----Validamos nombre de ususario y contraseña-----
function hacerLogin() {

```

```

let nombreUsuario = document.querySelector("#loginUsuario").value;
let clave = document.querySelector("#loginClave").value;
let mensaje = "";

let login = sistema.verificarLogin(nombreUsuario, clave);
if (login === true) {
    document.querySelector("#navPrincipal").style.display = "block";
    document.querySelector("#loginUsuario").value = "";
    document.querySelector("#loginClave").value = "";
    usuarioLogueado = sistema.obtenerObjeto(sistema.censistas,
"nombreUsuario", nombreUsuario);
    mostrarUsuarioLogueado();
} else {
    mensaje = "Usuario y/o contraseña incorrectos";
}
document.querySelector("#mensajeLogin").innerHTML = mensaje;
}

function logout() {
    alert("Gracias por su registro")
    document.querySelector("#seccionInfoEstadisitca").style.display =
"none";
    document.querySelector("#seccionReasignar").style.display = "none";
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none";
    document.querySelector("#seccionLogin").style.display = "none";
    document.querySelector("#navPrincipal").style.display = "none";
    document.querySelector("#seccionRegistrarCenso").style.display =
"none";
    document.querySelector("#seccionBuscador").style.display = "none";
    document.querySelector("#navPrincipal").style.display = "none";
    document.querySelector("#seccionReasignar").style.display = "none";
    document.querySelector("#nombreUsuarioLogueado").innerHTML = ""
    document.querySelector("#seccionPerfiles").style.display = "block";
    document.querySelector("#seccionInvitados").style.display = "none";
    usuarioLogueado = null
}

function cargarSlcDpto() {
    document.querySelector("#slcDepartamento").innerHTML = `<option
value="-1">Seleccione una opcion...</option>`;
    for (let i = 0; i < sistema.departamentos.length; i++) {
        const unDpto = sistema.departamentos[i];
        document.querySelector("#slcDepartamento").innerHTML += `<option
value="${unDpto.nombre}">
        ${unDpto.nombre}</option>`;
    }
}

```

```

}
function cargarSlcOcup() {
    document.querySelector("#slcOcupacion").innerHTML = `<option value="-
1">Seleccione una opcion...</option>`;
    for (let i = 0; i < sistema.ocupaciones.length; i++) {
        const unaOcup = sistema.ocupaciones[i];
        document.querySelector("#slcOcupacion").innerHTML += `<option
value="${unaOcup.ocupacion}">
        ${unaOcup.ocupacion}</option>`;
    }
}

//aca mostrar formulario registro de usuarios Censistas
function mostrarRegistro() {
    document.querySelector("#seccionReasignar").style.display = "none";
    document.querySelector("#seccionRegistrarUsuario").style.display =
"block";
    document.querySelector("#seccionLogin").style.display = "none";
    document.querySelector("#seccionInfoEstadisitca").style.display =
"none";
}
let idCensista = 4
//funcion para registrar perfil censita
function registrarCensista() {
    let nombre = document.querySelector("#nombreCen").value;
    let nombreUsuario = document.querySelector("#nombreUsuario").value;
    let clave = document.querySelector("#clave").value;
    let mensaje = "";

    if (sistema.validarRegistroOk(nombre, nombreUsuario, clave)) {
        if (sistema.validarContrasena(clave)) {
            if (!sistema.buscarElemento(sistema.censistas,
"nombreUsuario", nombreUsuario)) {
                let censista = new Censista(nombre, nombreUsuario, clave,
id);

                sistema.agregarCensista(censista);
                idCensista++
                mensaje = "Registro con exito!"
                document.querySelector("#nombreCen").value = "";
                document.querySelector("#nombreUsuario").value = "";
                document.querySelector("#clave").value = "";
                document.querySelector("#navPrincipal").style.display =
"block"

                usuarioLogueado =
sistema.obtenerObjeto(sistema.censistas, "nombreUsuario", nombreUsuario)
                mostrarUsuarioLogueado();
            }
        }
    }
}

```

```

        else {
            mensaje = "Ya existe un usuario con ese nombre";
        }
    }
    else {
        mensaje = "Formato de contraseña incorrecto: La contraseña
deberá tener un mínimo de 5 caracteres, contando con al menos una
mayúscula, una minúscula y un número"
    }
}
else {
    mensaje = "Los campos no pueden ser vacíos";
}
document.querySelector("#mensajeRegistro").innerHTML = mensaje;
}

//-----Formulario registro censos
function CargarRegistroCenso() {
    document.querySelector("#seccionRegistrarCenso").style.display =
"block"
    document.querySelector("#seccionBuscador").style.display = "none"
    document.querySelector("#seccionReasignar").style.display = "none"
    document.querySelector("#seccionInfoEstadistica").style.display =
"none"
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none"
    document.querySelector("#btnGuardarInvitado").style.display = "none"
}
function guardarCenso() {
    let nombre = document.querySelector("#nombreRegistroCenso").value;
    let apellido =
document.querySelector("#apellidoRegistroCenso").value;
    let edad =
Number(document.querySelector("#edadRegistroCenso").value);
    let cedula =
document.querySelector("#cedulaRegistroCenso").value.toString();
    let depto = document.querySelector("#slcDepartamento").value;
    let ocup = document.querySelector("#slcOcupacion").value;
    let mensaje = "";
    let idCensista = usuarioLogueado;

    cedula = sistema.eliminarCaracter(cedula, '.');
    cedula = sistema.eliminarCaracter(cedula, '-');

```

```

        if (nombre != "" && apellido != "" && edad > 0 && edad < 130) {
            if (sistema.validarCedula(cedula)) {
                if (!sistema.buscarElemento(sistema.censados, "cedula",
cedula)) {
                    if (depto != -1 && depto != "" && ocup != -1 && ocup !=
"" ) {
                        let unCenso = new Censado(nombre, apellido, edad,
cedula, depto, ocup, idCensista, id);
                        estadoValidado = false
                        sistema.censados.push(unCenso);
                        id++;
                        mensaje = "Datos guardados correctamente";
                        document.querySelector("#nombreRegistroCenso").value
= "";
                        document.querySelector("#apellidoRegistroCenso").value = "";
                        document.querySelector("#edadRegistroCenso").value =
"";
                        document.querySelector("#cedulaRegistroCenso").value
= "";
                        document.querySelector("#slcDepartamento").value =
"";
                        document.querySelector("#slcOcupacion").value = "";
                    } else {
                        mensaje = "Debe seleccionar una opcion";
                    }
                } else {
                    mensaje = "C.I ya registrado";
                }
            } else {
                mensaje = "C.I no valida";
            }
        } else {
            mensaje = "Los campos nombre y apellido no deben estar vacios, y
la edad debe ser menor a 130";
        }
        document.querySelector("#mensajeRegistroCenso").innerHTML = mensaje;
    }
    function validarCenso() {
        let nombre = document.querySelector("#nombreRegistroCenso").value;
        let apellido =
document.querySelector("#apellidoRegistroCenso").value;
        let edad =
Number(document.querySelector("#edadRegistroCenso").value);
        let cedula = document.querySelector("#cedulaRegistroCenso").value;
        let depto = document.querySelector("#slcDepartamento").value;
        let ocup = document.querySelector("#slcOcupacion").value;

```

```

let mensaje = "";
let idCensista = usuarioLogueado

cedula = sistema.eliminarCaracter(cedula, '.');
cedula = sistema.eliminarCaracter(cedula, '-');

if (sistema.validarCedula(cedula)) {
    if (nombre !== "" && apellido !== "" && edad > 0 && edad < 130) {
        if (!sistema.buscarElemento(sistema.censados, "cedula",
cedula)) {
            if (depto != -1 && ocup != -1 && depto != "" && ocup !=
"" ) {
                let validado = new Censado(nombre, apellido, edad,
cedula, depto, ocup, idCensista, id, estadoValidado = true);
                validado.estadoValidado = true;
                id++;
                sistema.agregarCensados(validado);
                mensaje = "Datos validados";
                document.querySelector("#nombreRegistroCenso").value
= "";
document.querySelector("#apellidoRegistroCenso").value = "";
                document.querySelector("#edadRegistroCenso").value =
"";
                document.querySelector("#cedulaRegistroCenso").value
= "";
                document.querySelector("#slcDepartamento").value =
"";
                document.querySelector("#slcOcupacion").value = "";
            } else {
                mensaje = "Debe seleccionar una ocupacion y/o
departamento";
            }
        } else {
            mensaje = "C.I ya registrada";
        }
    } else {
        mensaje = "Los campos no pueden estar vacios";
    }
} else {
    mensaje = "C.I no valida";
}
document.querySelector("#mensajeRegistroCenso").innerHTML = mensaje;
}

function validarEnBusquedaCensista() {
    let cedulaBuscar = document.querySelector("#buscarCedula").value;
    cedulaBuscar = sistema.eliminarCaracter(cedulaBuscar, '.');

```

```

        cedulaBuscar = sistema.eliminarCaracter(cedulaBuscar, '-');
        let validar = sistema.obtenerObjeto(sistema.censados, "cedula",
cedulaBuscar)
        if (validar !== null) {
            validar.estadoValidado = true;
            document.querySelector("#pResultadoBusqueda").innerHTML =
"Registro validado con éxito!";
            document.querySelector("#btnValidarCensos").style.display =
"none";
        } else {
            document.querySelector("#pResultadoBusqueda").innerHTML = "No se
encuentra ese registro";
            document.querySelector("#buscarCedula").innerHTML = "";
        }
    }
}
//variable global para los id de las precargas, comienza en 31 así los
nuevos reistros no toman un valor menor a 31.
id = 31
function guardarRegistroInvitados() {
    let nombre = document.querySelector("#nombreRegistroCenso").value;
    let apellido =
document.querySelector("#apellidoRegistroCenso").value;
    let edad =
Number(document.querySelector("#edadRegistroCenso").value);
    let cedula =
document.querySelector("#cedulaRegistroCenso").value.toString();
    let depto = document.querySelector("#slcDepartamento").value;
    let ocup = document.querySelector("#slcOcupacion").value;
    let mensaje = "";
    let idCensista = sistema.aleatorioID(sistema.censistas)
    let reas = sistema.obtenerObjeto(sistema.censistas, "id", idCensista)
    cedula = sistema.eliminarCaracter(cedula, '.');
    cedula = sistema.eliminarCaracter(cedula, '-');

    if (nombre !== "" && apellido !== "" && edad > 0 && edad < 130) {
        if (sistema.validarCedula(cedula)) {
            if (!sistema.buscarElemento(sistema.censados, "cedula",
cedula)) {
                if (depto !== -1 && depto !== "" && ocup !== -1 && ocup !==
"") {
                    let unCenso = new Censado(nombre, apellido, edad,
cedula, depto, ocup, idCensista, id);
                    estadoValidado = false
                    sistema.censados.push(unCenso);
                    id++;
                    mensaje = "Datos guardados correctamente";

```

```

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

document.querySelector("#apellidoRegistroCenso").value = "";
        document.querySelector("#edadRegistroCenso").value =
"";
        document.querySelector("#cedulaRegistroCenso").value
= "";
        document.querySelector("#slcDepartamento").value =
"";
        document.querySelector("#slcOcupacion").value = "";

document.querySelector("#seccionInvitados").style.display = "block"

document.querySelector("#seccionRegistrarCenso").style.display = "none"
        alert("Datos Guardado Correctamente - Censista
asignado = " + reas.nombre);
    } else {
        mensaje = "Debe seleccionar una opcion";
    }
    } else {
        mensaje = "C.I ya registrado";
    }
    } else {
        mensaje = "C.I no valida";
    }
    } else {
        mensaje = "Los campos nombre y apellido no deben estar vacios, y
la edad debe ser menor a 130";
    }
    document.querySelector("#mensajeRegistroCenso").innerHTML = mensaje;
}
//-----Busqueda censos-----
function cargarBuscador() {
    document.querySelector("#seccionBuscador").style.display = "block"
    document.querySelector("#btnValidarCensos").style.display = "none"
    document.querySelector("#seccionReasignar").style.display = "none"
    document.querySelector("#seccionRegistrarCenso").style.display =
"none"
    document.querySelector("#seccionInfoEstadisitca").style.display =
"none"
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none"
}
function buscarCensado() {
    let cedulaBuscar = document.querySelector("#buscarCedula").value

```



```

let cedulaLimpia = cedulaBuscar.replace(/[-.]/g, "");
let ci = null;

for (let i = 0; i < sistema.censados.length; i++) {
    const censado = sistema.censados[i];
    const cedulaSinFormato = censado.cedula.toString().replace(/[-.]/g, "");

    if (cedulaSinFormato === cedulaLimpia) {
        ci = censado;
        break;
    }
}

if (ci === null) {
    document.querySelector("#pResultadoBusqueda").innerHTML = "C.I no encontrado";
    document.querySelector("#btnValidarCenso").style.display = "none";
} else if (ci.estadoValidado === true) {
    document.querySelector("#pResultadoBusqueda").innerHTML = "C.I Registrado";
} else if (ci.estadoValidado === false) {
    document.querySelector("#pResultadoBusqueda").innerHTML = `
Nombre: <input type="text" value = "${ci.nombre}"> <br>
Apellido: <input type="text" value = "${ci.apellido}"> <br>
Edad: <input type="text" value = "${ci.edad}"> <br>
Departamento: <input type="text" value = "${ci.depto}"> <br>
Ocupacion: <input type="text" value = "${ci.ocupacion}">
`;
    document.querySelector("#btnValidarCensos").style.display = "block";
}
}

function buscarCiInvitado() {
    let cedulaBuscar = document.querySelector("#buscarCedulainv").value;
    //let ci = sistema.obtenerObjeto(sistema.censados, "cedula", cedulaBuscar);
    let cedulaLimpia = cedulaBuscar.replace(/[-.]/g, "");
    let ci = null;

    for (let i = 0; i < sistema.censados.length; i++) {
        const censado = sistema.censados[i];
        const cedulaSinFormato = censado.cedula.toString().replace(/[-.]/g, "");

        if (cedulaSinFormato === cedulaLimpia) {
            ci = censado;

```

```

        break;
    }
}
if (ci === null) {
    document.querySelector("#pResultadoBusquedaInv").innerHTML = "C.I no encontrado";
} else if (ci.estadoValidado === true) {
    document.querySelector("#pResultadoBusquedaInv").innerHTML = "C.I Registrado";
} else if (ci.estadoValidado === false) {
    document.querySelector("#btnGuardarCensosInvi").style.display = "none";
    for (let i = 0; i < sistema.censados.length; i++) {
        document.querySelector("#pResultadoBusquedaInv").innerHTML =
        <ul>
        <li>Nombre: ${ci.nombre} </li><br>
        <li>Apellido: ${ci.apellido}</li> <br>
        <li>Edad: ${ci.edad}</li> <br>
        <li>Departamento: ${ci.depto}</li> <br>
        <li>Ocupacion: ${ci.ocupacion}</li>
        </ul> <input type="button" value="Eliminar" data-
        invitado="${sistema.censados[i].id}" class="btnEliminar">
        ,
        let btnEliminarInvitado =
document.querySelectorAll(".btnEliminar");
        for (let i = 0; i < btnEliminarInvitado.length; i++) {
            btnEliminarInvitado[i].addEventListener("click",
eliminarInvitado); //eliminarInvitado funcion a crear
        }
document.querySelector("#btnMostrarEstadistica").style.display = "none";
    }
}
}

//-----Perfil Invitado formulario registro invitado-----
function seccionRegistrarInvi() {
    document.querySelector("#seccionRegistrarCenso").style.display =
"block"
    document.querySelector("#btnValidarCenso").style.display = "none"
    document.querySelector("#btnRegistrarCenso").style.display = "none"
    document.querySelector("#seccionInvitados").style.display = "none"
}

//-----Reasignar Censos-----

```

```

function mostrarSeccionReasignarCenso() {
    document.querySelector("#seccionReasignar").style.display = "block";
    document.querySelector("#seccionBuscador").style.display = "none";
    document.querySelector("#seccionRegistrarCenso").style.display =
"none";
    document.querySelector("#seccionInfoEstadisitca").style.display =
"none";
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none";
    document.querySelector("#slcReasignar").innerHTML = `<option value="-
1">Seleccionar</option>`;
    for (let i = 0; i < sistema.censados.length; i++) {
        let unaReasignacion = sistema.censados[i];
        if (unaReasignacion.estadoValidado !== true)
            document.querySelector("#slcReasignar").innerHTML += `<option
value="${unaReasignacion.id}">
                ${unaReasignacion.cedula}-
                ${unaReasignacion.nombre}
            </option>`;
    }
    document.querySelector("#slcCensistas").innerHTML = `<option value="-
1">Seleccionar</option>`;
    for (let i = 0; i < sistema.censistas.length; i++) {
        let unCensista = sistema.censistas[i];
        document.querySelector("#slcCensistas").innerHTML += `<option
value="${unCensista.id}">
                ${unCensista.nombre}-
                ${unCensista.nombreUsuario}-
            </option>`;
    }
}

function reasignarCenso() {
    let valorCensista =
Number(document.querySelector("#slcCensistas").value);
    let valorCenso =
Number(document.querySelector("#slcReasignar").value);

    let censado = sistema.obtenerObjeto(sistema.censados, "id",
valorCenso);
    let censista = sistema.obtenerObjeto(sistema.censistas, "id",
valorCensista);

    if (censado && censista) {
        censado.idCensista = censista.id;
        document.querySelector("#pReasignarPersona").innerHTML = "Censo
reasignado correctamente.";
    } else {

```

```

        document.querySelector("#pReasignarPersona").innerHTML = "No se
encontró el censado o el censista.";
    }
}
//-----Estadísticas-----
function mostrarSeccionEstadisticas() {
    document.querySelector("#seccionInfoEstadistitca").style.display =
"block"
    document.querySelector("#seccionReasignar").style.display = "none"
    document.querySelector("#seccionBuscador").style.display = "none"
    document.querySelector("#seccionRegistrarCenso").style.display =
"none"
    document.querySelector("#seccionRegistrarUsuario").style.display =
"none"
    document.querySelector("#slcSeleccionarDepto2").style.display =
"none"
    document.querySelector("#btnMostrarEstadisticas").style.display =
"none"
}
function totalPersonasCensadas() {
    let censosTrue = 0;
    for (let i = 0; i < sistema.censados.length; i++) {
        let mostrar = sistema.censados[i]
        if (mostrar.estadoValidado === true)
            censosTrue++
    }
    document.querySelector("#pTotalCensados").innerHTML = censosTrue
}

function censadosPorDepto() {
    let ver = sistema.contarPersonasPorDepartamento(sistema.censados,
sistema.departamentos)
    mostrarTablaConteo(ver);
}

function mostrarTablaConteo(conteo) {
    var tabla = '<table>';

    // Encabezado de la tabla
    tabla += '<tr><th>Departamento</th><th>Cantidad de personas
censadas</th></tr>';

    // Filas de la tabla con los datos del conteo
    for (var depto in conteo) {
        tabla += '<tr><td>' + depto + '</td><td>' + conteo[depto] +
'</td></tr>';
    }
}

```

```

    tabla += '</table>';

    // Mostrar la tabla en el elemento con el ID "tabla-conteo"
    document.getElementById('tabla-conteo').innerHTML = tabla;
}

function personasPendientes() {
    let totalCensados = sistema.censados.length
    let totalPendientes = 0
    for (let i = 0; i < sistema.censados.length; i++) {
        let censo = sistema.censados[i];
        if (censo.estadoValidado !== true) {
            totalPendientes++
        }
    }
    let final = ((totalPendientes / totalCensados) * 100).toFixed(2)

    document.querySelector("#pTotalpendientes").innerHTML = final + "%"
}

//-----Estadistica por departamento-----
function seleccionarDeptosTotal() {
    document.querySelector("#slcSeleccionarDepto2").style.display =
"block"
    document.querySelector("#btnVerDeptos").style.display = "none"
    document.querySelector("#btnMostrarEstadisticas").style.display =
"block"
    let slc = '<select id="slcSeleccionarDepto2">'
    slc += '<option value="-1">Seleccione una opcion...</option>'
    for (let i = 0; i < sistema.departamentos.length; i++) {
        let unDepto = sistema.departamentos[i]
        slc += `<option
value="${unDepto.nombre}">${unDepto.nombre}</option>`
    }
    slc += '</select>'
    document.getElementById('slcSeleccionarDepto2').innerHTML = slc

    document.querySelector("#btnMostrarEstadisticas").addEventListener("click
", mostrarEstadisticasPorDepartamento);
}

function mostrarEstadisticasPorDepartamento() {
    let selectElement = document.getElementById("slcSeleccionarDepto2");
    let departamentoSeleccionado = selectElement.value;

    if (departamentoSeleccionado === "-1") {
        document.querySelector("#pMostrarPorDepto").innerHTML = "";
    }
}

```

```

        return;
    }

    var estadisticas =
sistema.contarEdadPorDepartamento(sistema.censados,
sistema.departamentos, departamentoSeleccionado);

    // Crear la cadena de texto para mostrar en el elemento HTML
    var textoMostrar = "Cantidad de personas mayores de edad: " +
estadisticas.cantidadMayores + "<br>";
    textoMostrar += "Cantidad de personas menores de edad: " +
estadisticas.cantidadMenores;

    // Mostrar el resultado en el elemento HTML
    document.querySelector("#pMostrarPorDepto").innerHTML = textoMostrar;
}

function eliminarInvitado() {
    // let idInvitadoElimar = Number(this.getAttribute("data-invitado"));
    let idInvitadoElimar =
Number(document.querySelector("#buscarCedulainv").value);

    let objetoInvitadoEliminar = sistema.obtenerObjeto(sistema.censados,
"cedula", idInvitadoElimar);

    let confirmarEliminar = confirm(`¿Desea eliminar los datos
${objetoInvitadoEliminar.nombre}?`)
    if (confirmarEliminar) {

        for (let i = 0; i < sistema.censados.length; i++) {
            if (sistema.censados[i].cedula === idInvitadoElimar) {
                sistema.eliminarCenso(i);
                break;
            }
        }

        document.querySelector("#pResultadoBusquedainv").style.display =
""
        document.querySelector("#pResultadoBusquedainv").innerHTML =
`Registro eliminado: ${objetoInvitadoEliminar.nombre}`
    }
}

function mostrarTablaConteoInvitados(conteo) {
    var tabla = '<table>';

    // Encabezado de la tabla

```

```

    tabla += '<tr><th>Departamento</th><th>Estudian</th><th>No
trabajan</th><th>Dependientes o Independientes</th><th>Porcentaje del
total de censados</th></tr>';

    // Filas de la tabla con los datos del conteo
    for (var depto in conteo) {
        var estadisticas = conteo[depto];
        tabla += '<tr>';
        tabla += '<td>' + depto + '</td>';
        tabla += '<td>' + estadisticas.estudiantes + '</td>';
        tabla += '<td>' + estadisticas.noTrabajan + '</td>';
        tabla += '<td>' + estadisticas.dependienteOIndependiente +
'</td>';
        tabla += '<td>' + estadisticas.porcentaje + '</td>';
        tabla += '</tr>';
    }

    tabla += '</table>';

    // Mostrar la tabla en el elemento con el ID "tablaInvitado"
    document.getElementById('tablaInvitado').innerHTML = tabla;
}

// Info Estadística del invitado
function tablaEstadisticaInvitado() {
    let llamar = sistema.contarPersonasPorDepartamentoInvi()
    mostrarTablaConteoInvitados(llamar)
}

```

4.6 Código Clases.js

```

class Censista {
    constructor(nombre, nombreUsuario, contrasena, id) {
        this.nombre = nombre;
        this.nombreUsuario = nombreUsuario;
        this.contrasena = contrasena;
        this.id = id;
    }
}

class Censado {
    constructor(nombreCen, apellidoCen, edadCen, cedulaCen, deptoCen,
ocupCen, idCensista, idCen, estadoValidadoCen = false) {
        this.nombre = nombreCen;
        this.apellido = apellidoCen;
    }
}

```

```

        this.edad = edadCen;
        this.cedula = cedulaCen;
        this.depto = deptoCen;
        this.ocupacion = ocupCen;
        this.idCensista = idCensista;
        this.id = idCen;
        this.estadoValidado = estadoValidadoCen;
    }
}

class Departamento{
    constructor(idDepartamento, departamentoNombre){
        this.id = idDepartamento;
        this.nombre = departamentoNombre;
    }
}

class Ocupacion{
    constructor(idOcupacion, ocuNombre){
        this.id = idOcupacion;
        this.ocupacion = ocuNombre;
    }
}

```

4.7 Código 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: #48B9A9;
}

h1,
h2,
h3,
h5,
h6 {
  font-family: 'Roboto Flex', sans-serif;
}

h2{
  width:60%;
  float:left;
}

h4{
  float:right;
  font-weight:400;
  width:30%;
  margin-top:26px;
  text-align:right;
}

table, th, td{
  border-collapse:collapse;
  border: #000 1px solid;
  padding:5px;
  background-color:#fff;
}

table img, td>input[type=button]{
  width:100px;

```

```

    min-width:50px;
}

th{
    background-color:#303036;
    color:#fff;
    font-weight:normal;
}

#aux a {
    color: #0e1c20;
}

#contenedor {
    width: 800px;
    background-color: #6C66AC;
    margin: 0 auto;
}

label{
    min-width: 80px;
    text-align:left;
    display:inline-block;
    vertical-align:top;
}

.duracion{
    float:right;
}

textarea{
    min-height:80px;
}

p.ver-mas {
    text-align: center;
    cursor: pointer;
    font-weight: 700;
    color: #FC5130;
}

#logo {
    margin: 20px;
    float: left;
    width: 100px;
    float: left;
}

```

```

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

#navPrincipal a {
  color: #FFFAFF;
  cursor: pointer;
  color: #fff;
  font-size: 14px;
}

#aux a {
  text-decoration: underline;
  cursor: pointer;
  color: #FFFAFF;
  font-size: 14px;
}

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

#navPrincipal li {
  font-family: 'Roboto Flex', sans-serif;
  list-style-type: none;
  padding: 10px;
  border-bottom: 2px solid #FFFAFF;
  margin: 5px;
  float: left;
}

#cabezal {
  background-color: #303036;
}

#contenido {
  /*background-color: #DFDFDF;*/
  /*width: 590px;*/
  padding: 20px;
  float: left;
}

```

```

}

#contenido img{
  max-width:100%;
  margin-bottom: 1%;
}

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

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

#pie {
  background-color: #303036;
  clear: both;
  padding: 20px;
  color: #fff;
  font-size: 12px;
}

input,
select, textarea {
  background-color: #303036;
  color: #FFFAFF;
  border: 1px solid #ccc;
  padding: 5px 10px;
  font-family: 'Roboto Flex', sans-serif;
  min-width: 200px;
  border-radius: 5px;
  font-size:14px;
}

input[type="button"] {
  background-color: #303036;
  color: #fff;
}

#pie a {
  text-decoration: none;
  color: #fff;
}

```

```
}  
  
input[type="button"]:disabled {  
  background-color: #9696a1;  
}
```