

Los textos del mismo color hacen referencia a que el valor de referencia o id es el mismo

INSERCIÓN DE TRAMITES

STORE: **strGenerarTramite**

- INPUT:
 - ID_AREA_DEL_SOLICITANTE INT
 - ID_TRAMITE_A_SOLICITAR INT (LLAVE PRIMARIA DE TABLA CAT_TRAMITE_GENERAL, indica el nombre del trámite y permite conocer su flujo de aprobación, si no lo tienes insértalo desde la vista -cat_tramites>nuevo tramite)
 - ID_USUARIO_SISTEMA_DEL_SOLICITANTE INT
- OUTPUT:
 - BANDER ALERT SI EL REGISTRO ES EXITOSO = 1 | ALIAS: (alert)
 - ID_TRAMITE_GENERAL_SOLICITUD LLAVE PRIMARIA DE LA TABLA TRAMITE GENERAL EN LA CUAL SE REGISTRO EL TRAMITE | ALIAS: (id_tramite_general_solicitud)
 - NUMERO(FOLIO) DEL TRAMITE | ALIAS: (numero_tramite)

DETALLE DE TRAMITES

STORE: **strReadTramiteDetalles**

- INPUT:
 - ID_TRAMITE_GENERAL_SOLICITUD (obtenido al generar el trámite) INT
- OUTPUT:
 - Consulta a necesidad del usuario:

```
CREATE DEFINER='avisppa'@'%' PROCEDURE `strReadTramiteDetalles` ( IN idTramite INT)
BEGIN
    DECLARE idTramiteGen INT;
    SET idTramiteGen = (SELECT id_tramite_general FROM tramite_general WHERE id_tramite_general_solicitud = idTramite);

    -- SELECCION DEL TIPO DE TRAMITE PARA FILTRAR DATOS SEGUN CASO
    IF (idTramiteGen = 1) THEN -- Permiso
        SELECT tg.id_tramite_general_solicitud, tg.id_tramite_general, tg.id_usuario_solicitante, tg.fecha_tramite,
            tg.numero_tramite, tg.detalle, tg.id_status_tramite, ctg.nombre_tramite,
            (SELECT CONCAT(u.nombre, ' ', u.paterno, ' ', u.materno) FROM usuarios u, usuario_sistema us WHERE us.id_usuario = u.id AND us.id_usuario_sistema = tg.id_u
            (SELECT g.name FROM usuario_sistema us, groups g WHERE us.id_grupo = g.id AND us.id_usuario_sistema = tg.id_usuario_solicitante) groupName,
            CONCAT('Del ',DATE_FORMAT(p.fecha_inicio, '%d-%m-%Y'),' al ',DATE_FORMAT(p.fecha_fin, '%d-%m-%Y')) periodoPermiso,
            cpa.tipo_permiso, p.id_permiso, p.diaparcial, p.descripcion_comentario, p.vigencia, p.aplica_asueto, p.estatus_cic, p.autorizado
        FROM cat_tramite_general ctg, tramite_general tg, cat_permisos_asueto cpa, permisos p
        WHERE ctg.id_tramite_general = tg.id_tramite_general AND p.id_tramite = tg.id_tramite_general_solicitud
        AND cpa.id_permiso_asueto = p.id_permiso_asueto
        AND tg.id_tramite_general_solicitud = idTramite;
    ELSEIF (idTramiteGen = 2) THEN -- Justificación de Incidencias
        SELECT tg.id_tramite_general_solicitud, tg.numero_tramite, tg.fecha_tramite, tg.detalle, ctg.nombre_tramite,
            concat(u.nombre, ' ', u.paterno, ' ', u.materno) AS informacion
        FROM cat_tramite_general ctg, tramite_general tg, usuarios u
        WHERE tg.id_tramite_general_solicitud = idTramite
        AND tg.id_tramite_general = ctg.id_tramite_general
```

Ejemplo

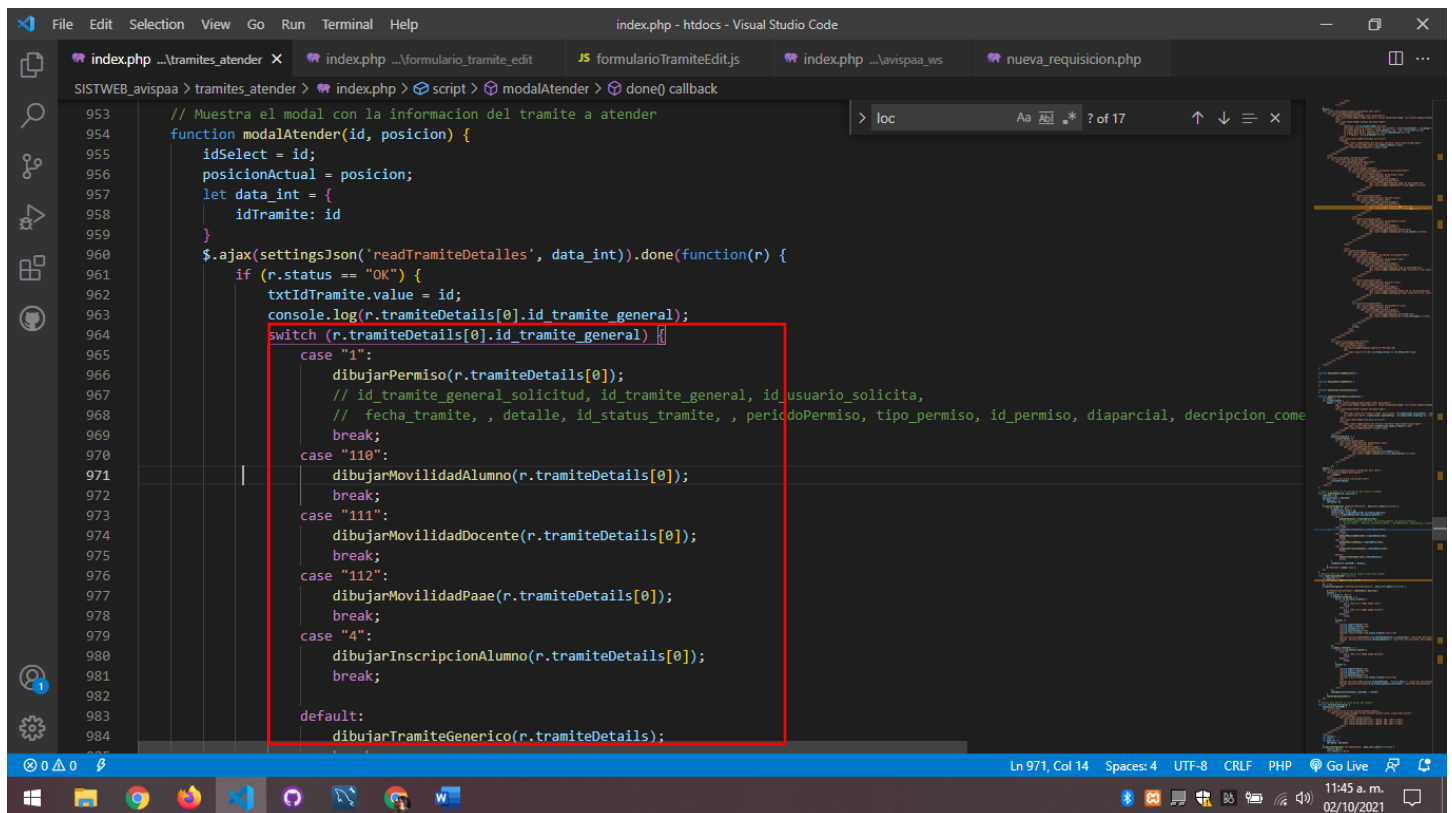
- Por medio del ID_TRAMITE_GENERAL_SOLICITUD(idTramite) del cual obtiene el ID_TRAMITE_GENERAL(idTramiteGen).
- El ID_TRAMITE_GENERAL permite la discriminación por casos para obtener el detalle necesario para cada uno de los tramites. Por lo que deberás integrar el caso de tu tramite por medio de su ID_TRAMITE_GENERAL para mostrar su detalle dentro de la vista TRAMITES_ATENDER.

DIBUJADO DE TRAMITE

VISTA: `tramites_atender`

Método: `modalAtender`

- INPUT:
 - `ID_TRAMITE_GENERAL_SOLICITUD` (id)
 - `POSICION` (referente a la posición que tiene para atender el trámite en el flujo, este valor puede ser 1, 0, o 2, si es 1 es el primero en el flujo, si es 0 es un intermedio en el flujo si es 2 es el último en el flujo).



```
953 // Muestra el modal con la informacion del tramite a atender
954 function modalAtender(id, posicion) {
955     idSelect = id;
956     posicionActual = posicion;
957     let data_int = {
958         idTramite: id
959     }
960     $.ajax(settingsJson('readTramiteDetalles', data_int)).done(function(r) {
961         if (r.status == "OK") {
962             txtIdTramite.value = id;
963             console.log(r.tramiteDetails[0].id_tramite_general);
964             switch (r.tramiteDetails[0].id_tramite_general) {
965                 case "1":
966                     dibujarPermiso(r.tramiteDetails[0]);
967                     // id_tramite_general_solicitud, id_tramite_general, id_usuario_solicita,
968                     // fecha_tramite, , detalle, id_status_tramite, , periodoPermiso, tipo_permiso, id_permiso, diaparcial, decripcion_come
969                     break;
970                 case "110":
971                     dibujarMovilidadAlumno(r.tramiteDetails[0]);
972                     break;
973                 case "111":
974                     dibujarMovilidadDocente(r.tramiteDetails[0]);
975                     break;
976                 case "112":
977                     dibujarMovilidadPaee(r.tramiteDetails[0]);
978                     break;
979                 case "4":
980                     dibujarInscripcionAlumno(r.tramiteDetails[0]);
981                     break;
982                 default:
983                     dibujarTramiteGenerico(r.tramiteDetails);
984             }
985         }
986     });
987 }
```

Este método permite el dibujo dinámico acorde al tipo de trámite que se recibe, por medio del consumo del store mostrado en DETALLE DE TRAMITES, deberas añadir el caso, acorde al numero de `ID_TRAMITE_GENERAL` correspondiente al tu trámite, al switch case y crear el método, que dibuje el detalle.

```

346
347
348 function dibujarMovilidadAlumno(e) {
349     console.log(e);
350     let tituloActividad2 = "";
351     let fila = "";
352     switch (e.actividad) {
353         case "1":
354             tituloActividad2 = "Área o laboratorio";
355             break;
356         case "2":
357             tituloActividad2 = "Título de ponencia";
358             break;
359         case "3":
360             tituloActividad2 = "Título de ponencia";
361             break;
362         case "4":
363             tituloActividad2 = "Actividad por otro evento";
364             break;
365         default:
366             tituloActividad2 = "Actividad por otro evento";
367             break;
368     }
369     if (e.actividad2 != "-") {
370         fila = `<li class="list-group-item">
371             <div class="todo-indicator bg-success"></div>
372             <div class="widget-content p-0">
373                 <div class="widget-content-wrapper">
374                     <div class="widget-content-left">
375                         <div class="widget-heading">${tituloActividad2}</div>
376                         <div class="widget-subheading"><span class="pr-2"><b class="text-primary">${e.actividad2}</b></span></div>
377                     </div>
378                 </div>
379             </li>`;
380     }
381     details = `
382     <div class="card-shadow-primary card-border mb-3 card">
383         <div class="dropdown-menu-header">
384             <div class="dropdown-menu-header-inner bg-primary">
385                 <div class="menu-header-image opacity-2" style="background-image: url('assets/images/dropdown-header/abstract1.jpg');">
386                 </div>
387                 <div class="menu-header-content btn-pane-right">
388                     <div>
389                         <h5>Folio: <b>${e.id_movilidad}</b></h5>
390                         <h5><span class="fa fa-user"></span> Solicitante: <b>${e.solicitante}</b> - <b>${e.grupo}</b></h5>
391                         <p class="pb-0 mb-0">Boleta: <b>${e.boleta}</b> </b> CVU: <b>${e.cvu}</b></p>
392                         <p class="pb-0 mb-0">Laboratorio: <b>${e.laboratorio}</b> </b> </p>
393                         <p>Programa: <b>${e.programa}</b></p>
394                     </div>
395                 </div>
396             </div>
397         </div>
398     `;
399 }

```

Dentro del método que dibuje podrás añadir las validaciones que requieras, pero este deberá otorgar un valor a la variable global **details** (no es necesario declararla dentro del método de dibujado), la cual tendrá el contenido y se podrá dibujar dentro del modal de atención.

ATENCIÓN DEL TRAMITE

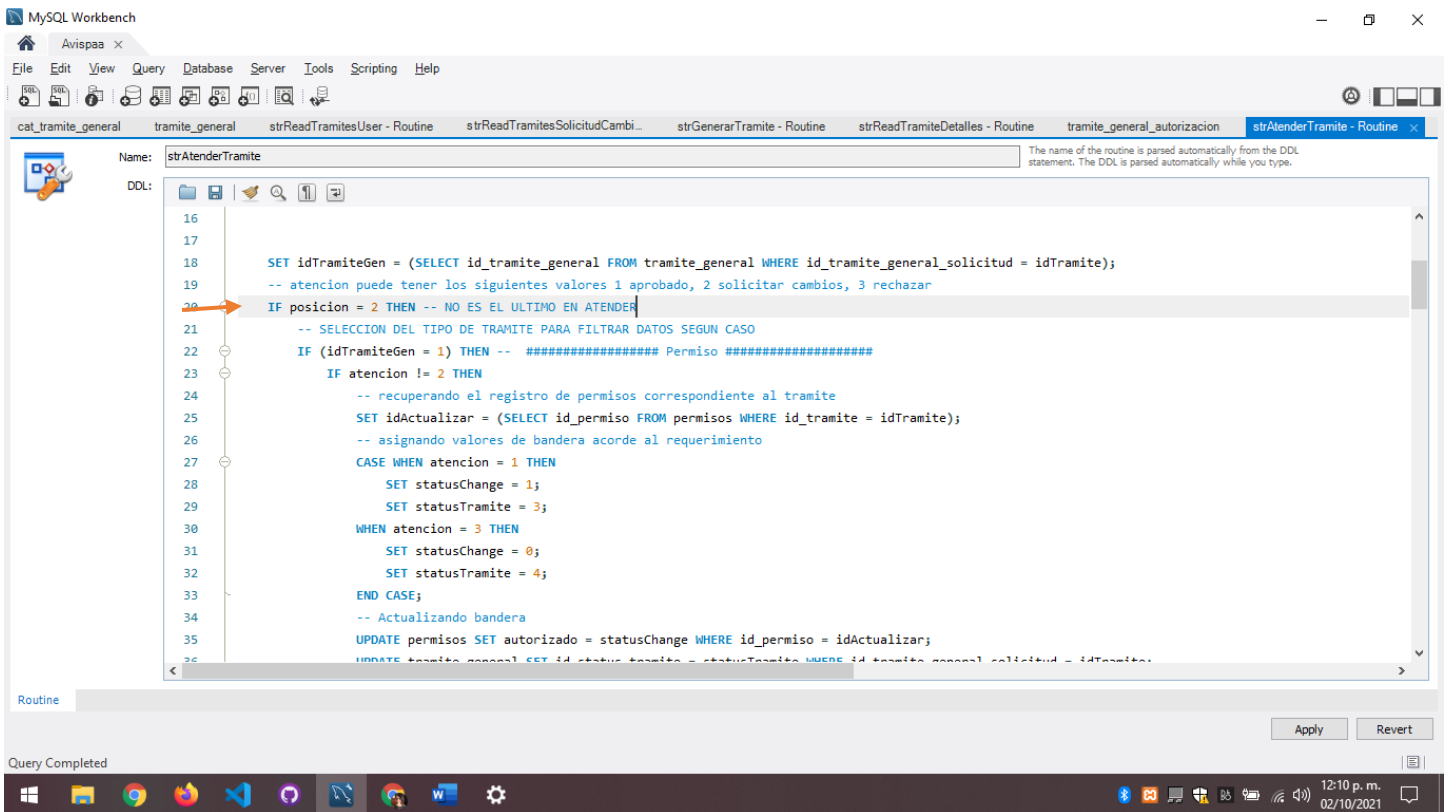
SERVICIO: **atenderTramite**

STORE: **strAtenderTramite**

- INPUT:
 - IN idTramite INT,
 - IN atencion INT,
 - IN idUsuario INT,
 - IN comentarioAtencion TEXT,
 - IN areaUsuario INT,
 - IN firma TEXT,
 - IN posicion INT
- OUTPUT:
 - Bandera: (alert) 1= exitoso, 0 = no atendido

Al igual que el store de detalle en base al tipo de tramite se desencadenan una serie de casos que realizaran la atención según corresponda, sin embargo, en este caso se valida primero la posición del usuario que está atendiendo el trámite, es decir

Si posición = 1, significa que el usuario es el primero en atender el trámite, 0 significa que es un intermedio en atender, y 2 es el ultimo en atender el trámite.



```
16
17
18 SET idTramiteGen = (SELECT id_tramite_general FROM tramite_general WHERE id_tramite_general_solicitud = idTramite);
19 -- atencion puede tener los siguientes valores 1 aprobado, 2 solicitar cambios, 3 rechazar
20 IF posicion = 2 THEN -- NO ES EL ULTIMO EN ATENDER
21 -- SELECCION DEL TIPO DE TRAMITE PARA FILTRAR DATOS SEGUN CASO
22 IF (idTramiteGen = 1) THEN -- ##### Permiso #####
23 IF atencion != 2 THEN
24 -- recuperando el registro de permisos correspondiente al tramite
25 SET idActualizar = (SELECT id_permiso FROM permisos WHERE id_tramite = idTramite);
26 -- asignando valores de bandera acorde al requerimiento
27 CASE WHEN atencion = 1 THEN
28 SET statusChange = 1;
29 SET statusTramite = 3;
30 WHEN atencion = 3 THEN
31 SET statusChange = 0;
32 SET statusTramite = 4;
33 END CASE;
34 -- Actualizando bandera
35 UPDATE permisos SET autorizado = statusChange WHERE id_permiso = idActualizar;
36 UPDATE tramite_general SET id_status_tramite = statusTramite WHERE id_tramite_general_solicitud = idTramite;
```

Como se observa en el IF, si es el último deberá revisar el tipo de tramite que se esta atendiendo y acorde a eso, realizar el update de estatus correspondiente en la tabla que reside el detalle y la tabla **tramite_general**.

Para estas acciones se tienen 2 variables que se pueden utilizar:

SET statusChange = 0; La cual puede residir el valor de la bandera para la tabla que contenga el detalle, el valor será según el manejo de la tabla.

SET statusTramite = 4; La cual recibirá el valor del status a manejar dentro de la tabla **tramite_general**, su valor puede ser, 3 -> aprobado o 4->rechazado o 5-> solicitud de cambios

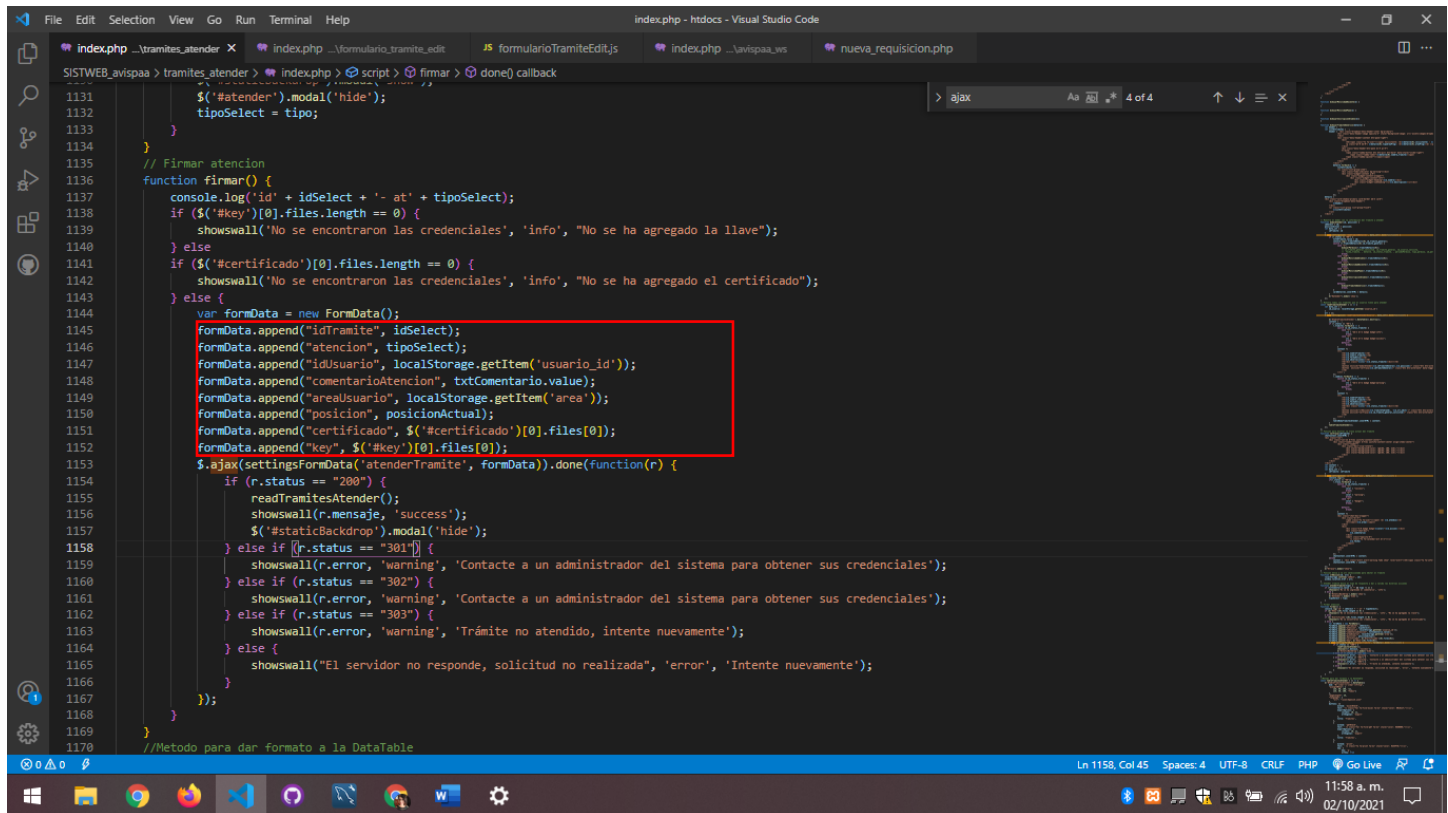
Las cuales realizaran la actualización de banderas de estatus en las tablas requeridas (tómese como ejemplo el trámite con id 1 permiso)

En caso de que sea el último, pero solicite cambios no se afectará la tabla que reside el detalle, por lo que solo se actualiza con la variable statusTramite en 5 para marcar el status de **tramite_general**.

En caso de que no sea el ultimo se realizaría una atención genérica.

Deberás incluir las banderas y tablas necesarias a actualizar acorde a tu caso de tipo de trámite.

Si tienes otra vista donde se requiera atender el trámite, deberás crear un método que obtenga los siguientes datos:



```
1131     $('#atender').modal('hide');
1132     tipoSelect = tipo;
1133 }
1134
1135 // Firmar atencion
1136 function firmar() {
1137     console.log('id' + idSelect + '-' + at + tipoSelect);
1138     if ($('#key')[0].files.length == 0) {
1139         showswall('No se encontraron las credenciales', 'info', "No se ha agregado la llave");
1140     } else
1141     if ($('#certificado')[0].files.length == 0) {
1142         showswall('No se encontraron las credenciales', 'info', "No se ha agregado el certificado");
1143     } else {
1144         var formData = new FormData();
1145         formData.append("idTramite", idSelect);
1146         formData.append("atencion", tipoSelect);
1147         formData.append("idUsuario", localStorage.getItem('usuario_id'));
1148         formData.append("comentarioAtencion", txtComentario.value);
1149         formData.append("areaUsuario", localStorage.getItem('area'));
1150         formData.append("posicion", posicionActual);
1151         formData.append("certificado", $('#certificado')[0].files[0]);
1152         formData.append("key", $('#key')[0].files[0]);
1153         $.ajax(settingsFormData('atenderTramite', formData)).done(function(r) {
1154             if (r.status == "200") {
1155                 readTramitesAtender();
1156                 showswall(r.mensaje, 'success');
1157                 $('#staticBackdrop').modal('hide');
1158             } else if (r.status == "301") {
1159                 showswall(r.error, 'warning', 'Contacte a un administrador del sistema para obtener sus credenciales');
1160             } else if (r.status == "302") {
1161                 showswall(r.error, 'warning', 'Contacte a un administrador del sistema para obtener sus credenciales');
1162             } else if (r.status == "303") {
1163                 showswall(r.error, 'warning', 'Trámite no atendido, intente nuevamente');
1164             } else {
1165                 showswall("El servidor no responde, solicitud no realizada", 'error', 'Intente nuevamente');
1166             }
1167         });
1168     }
1169 }
1170 //Metodo para dar formato a la DataTable
```

- SERVICIO: **atenderTramite**
- INPUT:
 - idTramite: ID_TRAMITE_GENERAL_SOLICITUD.
 - Atención: tiene 3 valores posibles, 1 aceptado, 2 solicitud de cambios, 3 rechazado.
 - idUsuario: ID_USUARIO_SISTEMA
 - comentarioAtencion: comentario ingresado por el usuario al atender el trámite.
 - areaUsuario: ID_AREA del usuario que atiende el trámite.
 - Posición: puede tener 3 valores 1, inicial en el flujo, 0 intermedio en el flujo, 2 final en el flujo.
 - Certificado y key para firma del trámite.
- OUTPUT:
 - Status 200: exitoso
 - Status 301: Usuario no certificado
 - Status 302: Las credenciales no corresponden al usuario
 - Status 303: atención no realizada
 - Status error: error de ejecución del servicio o BD.