PLEC DE PRESCRIPCIONS TÈCNIQUES

**SERVEI PER A L'ACTUALITZACIÓ DEL
PROGRAMA DEL SISTEMA DE GESTIÓ DE
LA FLOTA DE VEHICLES I DEL SERVEI DE
RECOLLIDA DE RESIDUS I NETEJA VIÀRIA
PER AL PROJECTE PILOT DE COMPRA
PÚBLICA INNOVADORA DEL CONSELL
COMARCAL DEL BAIX CAMP**

ÍNDEX

1	<u>ANTECEDENTS</u>	3
2	<u>INTRODUCCIÓ</u>	3
3	<u>OBJECTE</u>	4
4	<u>NOMBRE DE LOTS</u>	4
5	<u>PROCEDIMENT ESTABLERT</u>	5
5.1	CONTINGUT DE LES PROPOSTES	5
5.2	ABAST DELS TREBALLS	5
6	<u>ESPECIFICACIONS TÈCNIQUES</u>	5
6.1	GARANTIA	14
6.2	FORMACIÓ	14
6.3	TERMINI DE LLIURAMENT	14
7	<u>PRESSUPOST</u>	15

1 ANTECEDENTS

El Consell Comarcal del Baix Camp va crear l'any 1996 l'empresa Serveis Comarcals Mediambientals SA (SECOMSA) per a prestar el servei de recollida de residus de forma mancomunada en aquells municipis que li van delegar aquesta competència.

Actualment el Consell Comarcal del Baix Camp realitza a través de SECOMSA el servei de recollida de residus en 27 dels 28 municipis de la comarca, així com també a 3 municipis de l'Alt Camp.

SECOMSA disposa d'una flota de vehicles adaptada a les característiques del territori on presta el servei, amb capacitat per a la recollida selectiva de les fraccions resta, orgànica, paper, envasos, vidre, poda i els residus de deixalleria, independentment dels diferents sistemes de contenidors instal·lats a cada municipi.

A més a més de la plantilla i els equips destinats al servei de recollida de residus, SECOMSA disposa de personal administratiu per a la gestió de la informació i la facturació, i tècnics per assessorar als municipis i desenvolupar campanyes de comunicació i educació ambiental i projectes de nous serveis amb l'objectiu millorar els nivells de recollida selectiva.

SECOMSA també és la responsable de gestionar a planta de compostatge, l'estació de transferència i la deixalleria comarcal al Centre de Gestió de Residus (propietat del Consorci per a la Gestió dels Residus Municipals del Baix Camp), així com de les deixalleries de Cambrils i Mont-roig del Camp.

2 INTRODUCCIÓ

El març de 2020, el Ple del Consell Comarcal del Baix Camp va aprovar la sol·licitud per obtenir subvenció dels Fons Europeus de Desenvolupament Regional (FEDER) destinats a la realització de projectes pilot de compra pública d'innovació en matèria de recollida selectiva de residus municipals.

Entre els mesos de març i abril del 2020 la Comissió Europea va modificar els reglaments de desplegament dels fons europeus per oferir finançament a les administracions públiques i pal·liar així la despesa pública derivada de la pandèmia del coronavirus.

Per aquest motiu, aquesta convocatòria va passar a finançar-se amb fons propis de la Generalitat de Catalunya.

Mitjançant l'Acord del Govern, de 9 de febrer de 2021, s'aproven les despeses amb càrrec a pressupostos d'exercicis futurs de la partida de transferència a l'Agència de Residus de Catalunya del Departament de Territori i Sostenibilitat, per atendre la convocatòria de subvencions per a la realització de projectes pilot de compra pública d'innovació en matèria de recollida selectiva de residus municipals, per un import total de 10.000.000,00 d'euros.

En data 28 de juny de 2021 l'Agència de Residus de Catalunya va publicar l'atorgament de la subvenció sol·licitada pel Consell Comarcal del Baix Camp pel cofinançament de la realització del PROJECTE D'IMPLANTACIÓ DEL PAGAMENT PER GENERACIÓ ALS MUNICIPIS DE CAMBRILS, MONT-ROIG DEL CAMP I

VANDELLÒS I L'HOSPITALET DE L'INFANT - Projecte pilot de compra pública d'innovació en matèria de recollida selectiva de residus municipals.

La finalitat del projecte és augmentar tant la quantitat com la qualitat dels residus recollits selectivament. En el projecte es defineixen un seguit d'actuacions, a continuació es detallen les que són objecte del contracte:

L'actuació inclou la flota de vehicles de Secomsa que té instal·lada un sistema de gestió des del 31 de gener de 2019.

Així doncs, el present contracte inclou l'actualització del programari per a la seva gestió i implementació, així com tots la integració de tots els elements necessaris per a la correcta posada en funcionament.

3 OBJECTE

Constitueix l'objecte del present plec establir les condicions que regiran la contractació per al subministrament per a les actualitzacions del programa del sistema de gestió que té instal·lada la flota de vehicles que porta a terme el servei de recollida de residus i neteja viària.

Secomsa disposa d'una flota de vehicles per a la prestació del servei de recollida de residus que té integrada una plataforma de gestió de la flota de vehicles i dels servei de recollida de residus de residus i neteja viària que és de la marca Movertis i va ser adjudicat per Consell d'Administració del dia 17 de juliol de 2018. L'enllaç al perfil del contractant amb la formalització d'aquesta contractació, en el qual figuren les característiques tècniques del sistema de gestió de la flota, és la següent:

https://contractaciopublica.gencat.cat/ecofin_pscp/AppJava/awardnotice.pscp?idDoc=35990251&lawType=3&reqCode=viewDcan&idCap=10248555&

De totes maneres, es podrà concertar una visita, de manera potestativa, a les instal·lacions de SECOMSA, per poder resoldre dubtes tècnics respecte al funcionament o aspectes de la plataforma actual així com per a formular qualsevol altre aclariment.

Secomsa està implantant gràcies al cofinançament de l'Agència de Residus de Catalunya amb el projecte pilot de compra pública innovadora del Consell Comarcal del Baix Camp als municipis de Cambrils, Mont-roig del Camp i L'Hospitalet de l'Infant i Vandellòs un nou model de recollida.

L'objecte del contracte inclou el dret d'ús de les actualitzacions que apareguin, tant per a la incorporació de noves funcionalitats com per a solucionar errors o vulnerabilitats, a més de permetre disposar de noves versions de software objecte del manteniment.

4 NOMBRE DE LOTS

No es divideix en lots.

Atesa la naturalesa i l'objecte del contracte, no es considera adequat dividir-lo en lots, tal i com disposa l'article 99.3.b) de la Llei 9/2017, de 8 de novembre, de contractes del sector públic (LCSP), atès que la realització independent de les prestacions dificultaria

la correcta execució tècnica del contracte. Es considera que el fet de no dividir-lo en lots no restringeix injustificadament la competència d'empreses licitadores. Amb la qual cosa s'haurà de presentar oferta per a la totalitat de l'objecte del contracte.

5 PROCEDIMENT ESTABLERT

5.1 Contingut de les propostes

Per cada lot al que es vulgui optar, s'haurà d'entregar la següent documentació:

- **Oferta econòmica.** Màxim 25.000 euros més IVA vigent.
- **Termini de lliurament.** Màxim 3 mesos, és a dir, 90 dies naturals. L'oferta s'haurà de presentar en dies naturals.

5.2 Abast dels treballs

L'empresa adjudicatària haurà d'incloure al seu càrrec les següents activitats que formen part de l'objecte del contracte:

- Transport fins a la Planta de Compostatge.
- Proves fins assolir un correcte funcionament segons les especificacions tècniques.

6 ESPECIFICACIONES TÈCNIQUES

VALIDACIONES E INCIDENTES

Añadir nuevo campo de búsqueda por criterio municipio.

Menú de incidencias

Añadir opción escoger municipio tanto a abiertas como cerradas.

Para este requerimiento se deben aplicar los siguientes cambios:

- Se debe modificar la tabla de maestros para configurar el nuevo item del menú
- Se debe implementar la lógica requerida para el filtrado de las opciones del menú según los accesos otorgados y heredados de las parametrizaciones de roles y permisos
- Se debe implementar el nuevo permiso dentro de las paramétricas de permisos (los permisos deben confrontados según la autenticación basado en el token de acceso y el JWT administrado en sesión).
- Implementar la validación querida en el component y html del proyecto frontend.

Para el caso anterior se deberá cumplir las siguientes exigencias técnicas:

- Se deben crear las tablas relacionales y no relaciones según el modelo que permita almacenar en base de datos los registros necesarios asociados a las incidencias.
 - Fecha de la incidencia

- Nombre de la incidencia
 - Descripción de la incidencia
 - Tipo de incidencia
 - Usuario
 - Estado de la incidencia
- Se debe de implementar el modelo necesario dentro de la base de datos que permitan llevar el control y auditoría en relación a los eventos de creación, consulta, actualización y eliminación de una incidencia; lo anterior está basado en que se pueda realizar un seguimiento punto a punto del ciclo de vida de incidencias.
 - Fecha y hora del evento
 - Tipo de evento
 - Usuario que genera el eventoSe debe validar que la integridad del estado de la auditoría de incidencias concuerde con los registros de las incidencias registradas en las tablas propuestas en el modelo planteado.
 - Se deberán crear los servicios backend que permitan aplicar los criterios necesarios para cumplir los eventos de creación, consulta, actualización y eliminación de incidencias. Estos servicios deberán ser los responsables de establecer las conexiones necesarias entre las bases de datos y los componentes a nivel front.
Este componente debe cumplir con las reglas mencionadas a continuación
 - Validar que el registro concuerde con la geocerca configurada
 - Se debe manejar el cambio de estado de los incidentes
 - Se deberá implementar el formulario que permita el listar los incidentes, estos incidentes deberán permitir acceder al detalle de un incidente, esta pantalla debe ser alimentada desde el modelo especificado según la integración con base de datos.

VALIDACIONES Y AMONESTACIONES

Se debe crear un pantalla la cual permita lista todas las paradas no validadas de cada parte, donde conste chofer-peón-unidad-persona que ha validado y se debe poder crear desde esta pantalla automáticamente una carta de amonestación.

Este componente debe contar con los siguientes requerimientos:

- Se debe de crear las entidades correspondientes y la vista de consulta que permita mantener la indexación de las paradas a ser validadas por cada unidad
- Se debe crear API de consulta que integre los diferentes consumos de los servicios rest y micro servicios expuestos por el proveedor de administración de rutas, tareas y datos geoespaciales. Dicha integración de consultas debe realizarse de manera asíncrona cumplimiento con las restricciones de seguridad, concurrencia de sesiones indicadas por el proveedor.
- Estos llamados deben ser procesados por paquetes y en bloques de información según las restricciones de disponibilidad del servicio, tiempos de respuesta y límites de consultas.
- Se debe contemplar el uso de una cola de mensajería y manejo de cache de información que permita manejar el procesamiento de los datos de manera asíncrona garantizando el trazado de datos de las rutas y las tareas relacionadas a las mismas.

- Se debe contar con una tabla de auditoría que permita trazar los eventos capturando el dato del usuario en sesión; para esto se deben registrar estos eventos en las tablas de eventos.
- Se debe validar el tipo de rol y permiso del usuario en sesión con el fin de renderizar la información presentada en pantalla y las acciones que pueden ser realizadas sobre cada registro de paradas.
- Se debe actualizar la entidad backend y la tabla empleada para adicionar el campo de comentarios, este campos debe ser persistido y recuperado desde base de datos por cada registro.
- Permitir en la pantalla de parada colocar una anotación o comentario que permita justificar o realizar anotaciones sobre la parada; debe validarse el tipo de rol y los permisos sólo un rol superior puede cambiar el estado de un parte o paradas.
- visualizarse las paradas autorizadas y no autorizadas, se debe implementar el filtrado por medio de la implementación del tipo de estado en la tabla de paradas, se debe implementar la lógica que contraste el filtrado de los registros vs roles y permisos del usuario en sesión.
- El rol superior debe poder ruta, hora inicio, hora fin, nombre, plantilla, vehículo, columna de estado, comentario revisión anterior, link al mapa del sitio, contenedor, casilla para escribir notas.
- Generar la carta de amonestación. Para esto se debe implementar el componente frontend que permita al usuario generar la carta de amonestación según el formato indicado. De igual manera se requiere crear e implementar el servicio backend que ejecute la generación del documento.
- Se debe implementar a nivel de base de datos la table de auditoría para manejo de los históricos relacionados con las cortes de amonestación que han sido generadas
- Se debe contar con la funcionalidad que permita visualizar las carta de amonestación generada; para esto se debe mantener la integridad relacional entre el modelo de rutas, paradas y auditoría de cartas de amonestación
- Botón validar mostrarlo en la pantalla de paradas el cual permite actualizar el estado de validación de la parada en revisión.

VALIDAR PARTES

En relación a validar partes, se debe implementar las siguientes funcionalidades:

- Se debe incluir en la vista del listado de partes un filtro a nivel de la tabla "Filtro: No validado".
- Se debe implementar el desarrollo frontend que permita realizar el consumo del servicio backend a desarrollar, el cual contendrá la lógica para realizar la consulta de estados de partes hacia la base de datos.
- de implementará dentro del servicio backend existente, la nueva operación que permita realizar la consulta del estado de los partes
- Se deberá implementar en el frontend la lógica para obtener y procesar la respuesta dada por la nueva operación de consulta de estados de parte aplica al filtro.
- Se deberá implementar en el frontend la lógica que permita realizar el renderizado de la los registros de la tabla de partes, según el filtro aplicado,

PERMISOS Y USUARIOS

Pantalla para dar permisos a cada usuario, que menús puede acceder y de que empresas, y que sub-funciones en cada menú

- Se debe implementar una pantalla de administración de roles y permisos. Esta pantalla debe contar con las funcionalidades que permitan ejecutar las acciones de creación, eliminación, consulta y edición de los roles aplicables a la aplicación; por lo cual tendrá que crear las tablas en base de datos para la persistencia de las parametrización es de los roles.
- Se debe implementar una pantalla de administración de roles y permisos. Esta pantalla debe contar con las funcionalidades que permitan ejecutar las acciones de creación, eliminación, consulta y edición de los permisos aplicables a la aplicación; por lo cual tendrá que crear las tablas en base de datos para la persistencia de las parametrización es de los permisos.
- Se debe implementar la relación de rol/permiso para poder generar grupos de permisos aplicables a un rol específico, por lo cual debe contar se con las tablas requeridas que permitan parametrizar la asociación de roles/permisos.
- Se debe implementar la relación de rol/usuario para poder generar grupos de permisos aplicables a un usuario específico, por lo cual debe contar con las tablas requeridas que permitan parametrización la asoció de roles/usuario (esta relación es indispensable para permitir al frontend el renderizar correctamente las pantallas y funciones disponibilizar a los diferentes roles soportados en la aplicación).
- Se debe contar con una pantalla de consulta de roles y permisos. Esta pantalla debe contar con una tabla que permita listar los roles y permisos configurados en la aplicación
- Se debe contar con una pantalla de consulta de roles y usuarios. Esta pantalla debe contar con una tabla que permita listar por rol o usuario y que permita al usuario verificar que usuarios de la aplicación se encuentran asignados a un rol específico.
- Se debe modificar el servicio de autenticación; este servicio debe ser modificado para que permita desde el momento en el que usuario realiza su proceso de autenticación el cargar en sesión los permisos que el usuario tenga asociados según el rol asignado al mismo.
- Implementar la lógica para habilitar o deshabilitar funciones de la plataforma según los permisos asignados al usuario en sesión.

ACCESOS AYUNTAMIENTOS

Crear un acceso para ayuntamientos el cual estará limitado a unos listados específicos muy concretos donde la empresa deberá elegir que datos puede ver de cada punto recogido y que filtros permite aplicar siempre limitado únicamente a su Ayuntamiento y en modo de lectura.

- Se debe contar con la funcionalidad que permita la creación de usuarios según el ayuntamiento requerido.
- Se debe contar con la funcionalidad que permita listar los usuarios configurados en el sistema y que estos puedan ser filtrados de usuarios según el ayuntamiento requerido.
- Es necesario modificar el servicio de creación y edición de usuarios del sistema que permita adicionar el campo para identificar el ayuntamiento al que pertenece el usuario.
- Crear el esquema de base de datos para la parametrización de la relación usuario/ayuntamiento.
- Se debe modificar el servicio de autenticación para el manejo de renderizaciones. generales en todas y cada una de las vistas y componentes de la aplicación.
- Crear el esquema para la parametrización de funciones por roles y permisos.

- Implementar la lógica para habilitar o deshabilitar funciones de la plataforma según los permisos asignados al usuario en sesión.
- Se debe poder parametrizar el listado de ayuntamientos y usuarios pertenecientes a estos (manejo de grupos de usuarios por ayuntamientos).

ACTUALIZACIÓN APLICACIÓN

Debe suprimirse el módulo autoactualizable de la apk.

Es necesario aplicar los cambios en el archivo de configuración de compilación y despliegue de la aplicación móvil para desactivar la opción de actualización automática de la aplicación

FECHA/HORA

A partir de las 20:00 cambiamos al día siguiente, tanto al momento de abrir como de cerrar cualquier parte.

Para este caso se debe implementar:

Adicionar el campo fecha dentro de la tabla de registros de partes.

Se debe implementar la lógica backend que permita definir la fecha de inicio y finalización de los servicios (manejo de formato timestamp).

Se debe crear el componente frontend que permita ingresar la parametrización de la ficha de inicio y de finalización de los servicios.

Se debe implementar a nivel del modelo de datos, la tabla de parametrización de servicios que permita persistir y obtener los períodos de servicios parametrizados.

El sistema en la lógica anterior, podrá identificar todos los partes generados dentro del periodo de tiempo indicado contacto con la posibilidad de identificar que todos estos pertenecen al mismo día de ejecución.

CREAR CONTENEDOR (ELIMINAR FUNCIONALIDAD)

Es requerido eliminar de la aplicación todo el flujo correspondiente a la ejecución y eventos para la creación de contenedores (se debe afectar frontend y backend).

Para esto, es necesario:

- Eliminar la funcionalidad del html en el componente front.
- Eliminar el flujo backend frontend implementado para la funcionalidad mencionada.
- Eliminar la acción del servicio backend core encargado de la realización de la creación de contenedores.
- Eliminar de la aplicación toda la lógica asociada a la creación de contenedores.

EMPRESAS

- Es necesario implementar el control de perfiles empresariales, por lo cual se debe integrar el desarrollo de renderizar o de las funciones de la aplicación en asocio a los servicios de autenticaciones y a los servicios de roles y permisos.
- Se debe modificar el modelo de administración de perfiles actualmente administrado por la aplicación y complementar el modelo actual con las relaciones de tablas necesarias que permitan cumplir con la logia de perfilamiento por empresas.
- Se debe crear la parametrización de perfilado en la aplicación según empresa.
- Tenemos dos empresas y éstas son diferentes por todo. Quedan campos como las validaciones que no se diferencian y todo el mundo ve todo.

CAMBIAR COLOR CONTENEDORES

El contenedor que se muestra en el Tablet debe cambiar de color una vez recogido.

- Es necesario realizar el diagnóstico en el servicio de consulta de contenedores y que parte de la lógica implementada impide que se presente el cambio de color y estados de los contenedores.
- Se debe aplicar la solución respectiva según el requerimiento inicial de comportamiento de la aplicación.
- Se debe implementar la optimización sobre los servicios que permiten la identificación de estados de los contenedores.
- Se debe implementar en base de datos los eventos de trazabilidad de cambios de estados de los contenedores.

INFORMES

Deben poder pedirse informes de todos los contenedores de cualquier municipio sin ningún tipo de limitación temporal y obtener la respuesta en un tiempo inferior siempre a 30 segundos.

La aplicación debe contar con la posibilidad de generar los informes requeridos bajo el performance requerido (formatos pdf - Excel)

- Se debe realizar una actualización de las librerías y dependencias empleadas para la generación de exportación de informes.
- Actualizar la lógica de negocio implementada para la generación de informes, permitiendo optimizar el performance de la misma.
- Optimizar las consultas realizadas sobre las tablas de base de datos y pensar en la implementación de vistas de consultas con valores de índices para lograr un mejor rendimiento en los tiempos de impactos sobre la base de datos.

PUNTOS y RUTAS

Han de crearse dos tipos nuevos de puntos, por un lado, puntos de pesaje, en los cuales el trabajador entrará el pesaje, el cual quedará repartido en todos los contenedores con el mismo código de cliente, y por otro, los puntos de paso, los cuales se mostrarán en el mapa y cambiarán de estado/color al llegar a ellos sin ninguna otra interacción. Estos mismos puntos deberán validarse y poder modificarse a posteriori junto con el resto de eventos del parte.

- Se debe implementar la funcionalidad de parametrización de “puntos de paso” y no “puntos”.
 - Se debe implementar en las entidades necesarias en el modelo de datos que permita manejar la persistencia Puntos de paso y no paso.
 - Implementar el servicio backend que permita realizar el CRUD de puntos de paso y no paso.
 - Se debe implementar el desarrollo de los servicios frontend que permitan realizar el CRUD de puntos de paso y no paso.
 - Se debe implementar la lógica de negocio necesaria que permita identificar de manera automática los puntos que tengan los mismos CFF y repartir de manera autogenerada entre otros contenedores con las mismas características y que se encuentren dentro de la ruta.
- **Introducir pesos manualmente desde la app:**

- En la pantalla de visualización del contenedor en la ruta, se debe implementar el evento que permita dar clic sobre el icono del contenedor
- Al dar clic sobre el icono del contenedor, la aplicación deberá mostrar un formulario emergente el cual permite ingresar manualmente.
- El formulario muestra los datos básicos del contenedor(No editables) y el campo peso en blanco y disponible para ser digitado por el usuario.
- Se implementará como parámetro al servicio actual el campo peso manual que permita enviar y persistir el dato ingresado hasta la base de datos.
- Se debe implementar una tabla de auditoría que permita llevar un registro de históricos de pasos ingresados manualmente.
- Introducir pantalla para validar peso donde aparezcan los pesajes entrados en cada punto Este campo deberá modificarse también a posteriori.
- Se debe actualizar la tabla contenedores para permitir trazar y persistir en base de datos el valor del pesaje real.
- Se debe modificar los servicios frontend y backend que permitan realizar el envío de datos a ser persistidos en base de datos.
- Los servicios backend deben cumplir en la implementación que permita validar al momento de generarla ruta la redistribución y reasignación de contenedores por lo cual también debe ser modificado el servicio de rutas encargado de realizar la generación y optimización de rutas según la distribución geográfica de los contenedores y de su configuración CFF.

Al entrar dentro de una ruta para editarla, añadir un campo “search”, a su vez, este listado debe poder ordenarse.

Hay que poder añadir y quitar contenedores de la ruta sin el mapa. Al ver el listado de debajo del mapa, poder filtrar y borrar a la vez que poder buscar por municipio y área los contenedores de la base de datos y elegir cual queremos añadir.

- Es necesario analizar la posibilidad de implementar un buscador en la aplicación, este buscador debe ejecutar acción sobre el mapa de visualización de contenedores; para esto se requiere impactar el servicio de mapas y rutas y adicionar los eventos necesarios la realizar la búsqueda y filtrado de los contenedores (importante contemplar la implementación de manejo de cacheado de información para no impactar constantemente al servicios de mapas).
- Se debe implementar los eventos necesarios que permitan el realizar la eliminación de un contenedor dentro de una ruta, es decir, se debe modificar la lógica implementa en el servicio de generación de rutas para que permita retirar un contenedor luego que esta ya ha sido generada y realizar nuevamente el cálculo y generar la nueva ruta.
- Se debe implementar los eventos necesarios que permitan el realizar la adición de un contenedor dentro de una ruta, es decir, se debe modificar la lógica implementa en el servicio de generación de rutas para que permita añadir un contenedor luego que esta ya ha sido generada y realizar nuevamente el cálculo y generar la nueva ruta.(LOCALIZANDO EN DESPLEGABLE POR NOMBRE Y NO UNICAMENTE POR MAPA)

MAPA DE TRABAJO

- Añadir a la funcionalidad tanto a nivel backend y frontend (menú) encima del mismo mapavque permita filtrar el contenido según localización datos geográficos(Municipios), empresas, tipos contenedores entre otros filtros.

EDITAR CALENDARIO RUTAS

- Se deben modificar los servicios backend de generación de rutas y conductores e implementar las interfaces con los nuevos flags de eventos (Disponibilizar el botón/modo de edición)

CONTENEDORES

- Se debe implementar la lógica a nivel backend que permita obtener el detalle de la ruta(s) a las que se encuentra asociado un contenedor. de igual manera se debe modificar el modal frontend para que al momento de confirmar la eliminación, sea posible identificar la ruta al que este pertenece.
- Debe poder eliminarse de unas rutas en concreto y no solo de todas.

SEGUIMIENTO CONTENEDORES RECOGIDOS

- Actualizar el api de integración y el servicio de gps que permita generar el seguimiento en tiempo real sobre el recorrido de una unidad.

FICHADOR

Debe generar un nuevo servicio de autenticación y login, el cual tome como llave de ingreso al conductor, adicionalmente, se debe modificar el componente frond que retirar las validaciones asociadas a las tarjetas de los conductores y implementar la nueva lógica que valide el listado maestro de conductores que es recuperado por el servicio backend desde la base de datos (tabla conductores).

PARTES

EDICIÓN DE PARTES REALIZADOS

Para este caso se debe implementar:

- Adicionar el campo fecha dentro de la tabla de registros de partes.
- Se debe adicionar el campos fecha en el servicio backend para que pueda ser enviado y persistido a base de datos.
- Es necesario implementar a nivel visual el campo de fecha, el cual debe ser editable por el usuario y éste será el dato a enviar al servicio backend.
- Deben poder adicionarse contenedores manualmente dentro de un parte mientras se valida.

PARTES MANUALES

En el menú de la aplicación debe contar la nueva opción de "Parte manual", el cual nos permitirá pasar un parte que tengamos en papel al sistema, teniendo en cuenta los mismo datos que cuando se genera automáticamente para (fecha, ruta, contenedores, personal, peso, etc) . Para generar esta funcionalidad es necesario:

- Adicional al componente frontend del menú la opción de parte manual
- La nueva opción del menú debe ir ligada a lógica roles y permisos aplicada al usuario en sesión
- Se debe implementar al componente que permita ingresar la información del parte a ser ingresado

- El formulario debe contar con todos los campos o propiedades iguales a las que son generadas de manera automática en las partes de la versión actual productiva.
- Se debe crear la funcionalidad a nivel backend que persistir dentro del modelo de base de datos el nuevo parte y cumplir con la lógica de restricciones e integridad de la base de datos (relación unidad-ruta-contenedor).

PARTES EN CURSO

Botón ELIMINAR para poder borrar por completo un parto (así podemos limpiar los antiguos, las pruebas y los errores) sin que pase ningún dato de ese parto a ningún registro. (poner doble confirmación)

Para este caso se debe implementar:

- Se debe implementar en la vista donde se pueda visualizar en el listado de partes la opción de eliminar. La cual puede ser opción de múltiple selección.
- Se debe modificar el servicio backend frontend y backend core para soportar el evento de eliminación.
- Implementar la lógica necesaria (Delete a base de datos) para realizar la eliminación de los registros realizados.
- Se debe tener en cuenta el manejo de roles y permisos, ya que esta acción solo puede ser ejecutada por un rol superior.

INFORMES

- Informe pdf Eliminar campo "ayuntamiento de" y dejar sólo nombre cliente: Para esto se debe modificar el componente html y remover el campo del modelo empleado para el binding de información desde el backend frontend.
- Listado contenedores El link "ver", lleva al mapa del contenedor pero debe llevar al mapa de trabajo, ya que este mapa no permite "mover") que es el objetivo de este link. Se debe modificar la entidad contenedor en backend y a nivel del modelo front para adicionar la propiedad link del detalle el cual debe ejecutar la redirección a la pantalla de detalle del contenedor.

Listado facturación:

Falta que el número de contenedor en las pantallas listado facturación/contenedores, sea un link en el mapa de trabajo y no al mapa de muestra, para lo anterior se debe implementar los siguientes cambios:

- Se debe modificar la entidad contenedor en backend y a nivel del modelo front para adicionar la propiedad link del detalle de facturación.
- Se debe modificar el servicio backend para adicionar la lógica que permita generar el listado de facturación.
- Se debe implementar la redirección y renderizar o a la pantalla de facturación.

LISTADO RRHH

Se requiere contar con la pantalla y funcionalidades que permitan generar el listado y/o Dashboard de seguimiento de las horas de trabajo ejecutadas por un trabajador (chofer/peon). Teniendo en cuenta que deben poderse filtrar entre fechas, por nombre y por cada trabajo desempeñado, ya sea como chofer o peón. Para esta funcionalidad es necesario:

- Crear el modelo en base de datos necesarios que permita realizar el proceso de persistencia y consulta de horas de trabajo.
- Se debe crear el modelo que permita manejar la parametrización o modelo de jornadas (inicio/finalización de jornada).

- Se debe crear el servicio el cual implemente los eventos de trazabilidad y crud sobre la parametrización de jornadas.
- Se debe crear el servicio que permita recuperar la auditoría de horas trabajadas por trabajador.
- Se debe crear los componentes frontend que permitan realizar el consumo de los servicios backend para dejar el modelo con los datos necesarios a ser mostrados a nivel pantalla aplicando la lógica de relación trabajador/auditoría.
- El sistema debe permitir generar seguimiento por Jornada u horarios ejecutados por cada trabajador, para esto es necesario implementar el servicio de informe Trabajador para consultar la auditoría del trabajador y contrastar con los registros generados desde la aplicación.

INFORME AYUNTAMIENTOS:

Opción todos: generar un pdf por cada ayuntamiento por separado (ahora hace sólo todos juntos) y mandarlo automáticamente desde la aplicación por mail a cada ayuntamiento. Columnas levantadas en la tabla resumen Totales en las tablas de cada fracción tanto abajo como a la derecha. Añadir totalizaciones a las tablas generadas en todas las direcciones y poner título "aixecades", en la totalización derecha de las tablas de cada fracción.

Para cumplir con este requisito es necesario:

- Modificar el servicio de generación de reportes, con el fin de agregar a la firma del consumo el parámetro flag (identificador del ayuntamiento), esto para que permita realizar el filtro sobre la consulta realizada a la base de datos y que los datos consultados pertenecen al ayuntamiento solicitado.
- Modificar el modelo del objeto enviado al servicio backend que permita enviar como parámetro el identificador del ayuntamiento previamente parametrizado en base de datos.
- Se debe contar una parametrización de ayuntamientos (maestro de ayuntamientos) que permita dar identidad única a cada ayuntamiento que es indexado en el sistema.
- Implementar las funcionalidades frontend requeridas para la generación de los informes, al momento de dar clic en la generación del informe, se debe captar el parámetro de identificador del ayuntamiento, esto implementado el filtro ayuntamiento dentro del listado de ayuntamiento e incorporar al registro la opción de generar reporte.

6.1 Garantia

S'estableix una garantia mínima de 36 mesos.

6.2 Formació

Caldrà realitzar formació per a l'actualització del software o per les noves incorporacions de funcionalitats.

6.3 Termini de lliurament

El termini total de lliurament totes les actualitzacions i dels elements de l'objecte del contracte es fixa en un màxim 3 mesos, a comptar des de la data de formalització del contracte.

7 PRESSUPOST

El pressupost base d'aquesta contractació és de 25.000 euros més 5.250 euros d'IVA (21%). En total el pressupost base de licitació és de 30.250 euros.