

```

public with sharing class R1_CLS_MissionTriggerMethods {

    public static void insertVisionRecords (List<R1_Mission__c> misionNews){
        List<R1_Vision__c> relatedVisionslist = new List<R1_Vision__c>();

        Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_VISION =
        Schema.SObjectType.R1_Vision__c.getRecordTypeInfosByDeveloperName();
        Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_SERVICE_CONCEPT =
        Schema.SObjectType.R1_ConceptService__c.getRecordTypeInfosByDeveloperName();

        Id RECORDTYPE_DESTINATION_SERVICE_CONCEPT =
        getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_destinationConcept');
        Id RECORDTYPE_NORMAL_SERVICE_CONCEPT =
        getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_normalConcept');
        Id RECORDTYPE_EA_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
        'R1_VIS_RT_EAVision');
        Id RECORDTYPE_PROVIDER_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
        'R1_VIS_RT_ProvVision');

        Set<Id> setServicesProvided = new Set<Id>();
        Set<String> setTarifasVigentes = new Set<String>();
        Map<String, String> offerTariffPAMap = new Map<String, String>();
        System.debug('Lo de los tipos pesados');
        for(R1_Mission__c m : misionNews) {
            //////////////////////////////////////
            System.debug(m.R1_MIS_PIC_Type__c + ' ' + m.R1_MIS_PIC_Truck__c + ' ' +
            m.R1_MIS_TEX_Platnumber_trailer__c + ' ' + m.R1_MIS_TEX_Rear_Axles__c + ' ' +
            m.R1_MIS_NUM_Weight__c);
            //////////////////////////////////////

            setServicesProvided.add(m.R1_MIS_LOO_Services_Provided__c);
            System.debug(m.R1_MIS_LOO_Services_Provided__c);
        }
        List<R1_Offer_Tariff__c> tarifaVigente = [SELECT Id, R1_OFT_TEX_ID_Tariff__c,
        R2_OFT_LOO_OfferTarif__r.R1_OFT_TEX_ID_Tariff__c ,
        R1_OFT_LOO_Presta_Assistance__r.R1_SPR_LOO_PaymentInfo__r.R1_DFA_LOO_IVA__r.R1_RE
        G_PER_Quantity__c, R1_OFT_LOO_Presta_Assistance__c,
        R1_OFT_PIC_Service_Type__c,
        R1_OFT_LOO_Presta_Assistance__r.R1_SPR_TEX_Code__c,
        R1_OFT_LOO_Presta_Assistance__r.R1_SPR_PIC_ServiceType__c
        FROM R1_Offer_Tariff__c
        WHERE R1_OFT_LOO_Presta_Assistance__c IN :
        setServicesProvided];

        Map<Id, String> mapPAs = new Map<Id, String>();
        for(R1_Offer_Tariff__c offerTariff : tarifaVigente) {
            setTarifasVigentes.add(offerTariff.R2_OFT_LOO_OfferTarif__r.R1_OFT_TEX_ID_Tariff__c);
            offerTariffPAMap.put(offerTariff.R1_OFT_LOO_Presta_Assistance__c,
            offerTariff.R2_OFT_LOO_OfferTarif__r.R1_OFT_TEX_ID_Tariff__c);
            if(!mapPAs.containsKey(offerTariff.R1_OFT_LOO_Presta_Assistance__c)){
                mapPAs.put(offerTariff.R1_OFT_LOO_Presta_Assistance__c,
                offerTariff.R1_OFT_LOO_Presta_Assistance__r.R1_SPR_PIC_ServiceType__c);
            }
        }

        System.debug('Mapa PRESTA ASISTENCIA - TARIFA ID');
    }
}

```

```
System.debug(offerTariffPAMap);
```

```
List<R1_Tariff_Line__c> tariffLineBase = [SELECT
R1_LDT_LOO_Concept_Type__r.R1_PRI_FOR_ConcpetName__c,
R1_LDT_LOO_Concept_Type__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R1_TIP_TEX_Description__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_AxesNumber__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_Minimum_Weight__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_Maximum_weight__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_CHE_Complete__c,R1_LDT_L
OO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_CHE_ROS__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_PIC_Tipolgy__c,
R1_LDT_MAS_Offer_Tariff__r.R1_OFT_TEX_ID_Tariff__c,R1_LDT_LOO_Concept_Type__r.R1_PRI
LOO_Concept__r.R1_CON_TEX_ACENaMe__c,
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_NUM_MinKM__c,R1_LDT_L
OO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_NUM_MaxKm__c
FROM R1_Tariff_Line__c
WHERE
R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R1_TIP_PIC_Unit__c = 'Euro'
AND R1_LDT_MAS_Offer_Tariff__r.R1_OFT_TEX_ID_Tariff__c IN:
setTarifasVigentes];
```

```
String EAVisionId =
Schema.SObjectType.R1_Vision__c.getRecordTypeInfoByName().get('Vision
EA').getRecordTypeId();
String serviceConceptRT =
Schema.SObjectType.R1_ConceptService__c.getRecordTypeInfoByDeveloperName().get('R2_C
SE_RT_normalConcept').getRecordTypeId();

Map<String, List<R1_ConceptService__c>> mapaConceptos = new Map<String,
List<R1_ConceptService__c>>();

for(R1_Mission__c m : misionNews) {

    System.debug('PRESTA ASISTENCIA');
    // System.debug(m.R1_MIS_LOO_Services_Provided__r.R1_SPR_TEX_Code__c);
    System.debug(m.R1_MIS_LOO_Services_Provided__c);

    Map<String, R1_Tariff_Line__c> conceptPriceMap = new Map<String,
R1_Tariff_Line__c>();
    Map<String, Map<String,R1_Tariff_Line__c>> conceptPriceMapGP = new Map<String,
Map<String,R1_Tariff_Line__c>>();
    String tarifald = offerTariffPAMap.get(m.R1_MIS_LOO_Services_Provided__c);

    System.debug('TARIFA PADRE');
    System.debug(tarifald);
    Map<String, String> mapKeys = new Map<String, String>();

    for (R1_Tariff_Line__c tariffLine : tariffLineBase) {
        if (tariffLine.R1_LDT_MAS_Offer_Tariff__r.R1_OFT_TEX_ID_Tariff__c == tarifald){
            if(!
conceptPriceMapGP.containsKey(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Conce
pt__r.R1_CON_TEX_ACENaMe__c)){
                Map<String, R1_Tariff_Line__c> mapAux = new Map<String, R1_Tariff_Line__c>();

                mapAux.put(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R1_TIP_TEX_De
scription__c, tariffLine);
```

```

conceptPriceMapGP.put(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_
CON_TEX_ACENaMe__c, mapAux);
    }else{
        Map<String, R1_Tariff_Line__c> mapAux =
conceptPriceMapGP.get(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_
CON_TEX_ACENaMe__c);

mapAux.put(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R1_TIP_TEX_De
scription__c, tariffLine);

conceptPriceMapGP.put(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_
CON_TEX_ACENaMe__c, mapAux);
    }

conceptPriceMap.put(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CO
N_TEX_ACENaMe__c, tariffLine);
    }

if(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__
c == 'Urbano' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'URBANO' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Urbano (hasta 20km)'){
    if(!mapKeys.containsKey('Urbano')){
        mapKeys.put('Urbano',
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c)
;
    }
}

}else
if(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__
c == 'SALIDA' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Salida'){
    if(!mapKeys.containsKey('Salida')){
        mapKeys.put('Salida',
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c)
;
    }
}

}else
if(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__
c == 'Kilometro' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Kilometros' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Kilómetro' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'KILOMETROS'){
    if(!mapKeys.containsKey('Kilometro')){
        mapKeys.put('Kilometro',
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c)
;
    }
}

}else
if(tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__
c == 'Metrop(21-60 k)' ||

```

```

tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Metrop(21-40 k)' ||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'Metrop(21-50 k)'||
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c
== 'KILOMETROS'){
    if(!mapKeys.containsKey('Metro')){
        mapKeys.put('Metro',
tariffLine.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Concept__r.R1_CON_TEX_ACENaMe__c)
;
    }

}
}
}
}

```

System.debug('Mapa NOMBRE CONCEPTO - LINEA TARIFA');

```

String origin = m.R1_MIS_TEX_City_vehicleLocation__c; // Viene siempre
String base = m.R1_MIS_TEX_CityPA__c; // Viene siempre
String midpoint = m.R1_MIS_TEX_City_vehicleStop__c; // Puede no venir
String destiny = m.R1_MIS_TEX_City_vehicleDestination__c; // Puede no venir

```

```

Double eventToVehicleDestination = m.R1_MIS_NUM_eventToVehicleDestination__c !=
null? m.R1_MIS_NUM_eventToVehicleDestination__c:0;
Double intermediateStopToVehDestin =
m.R1_MIS_NUM_intermediateStopToVehDestin__c != null?
m.R1_MIS_NUM_intermediateStopToVehDestin__c:0;
Double intermediateStopToProvider = m.R1_MIS_NUM_intermediateStopToProvider__c !=
null? m.R1_MIS_NUM_intermediateStopToProvider__c:0;
Double eventToIntermediateStop = m.R1_MIS_NUM_eventToIntermediateStop__c != null?
m.R1_MIS_NUM_eventToIntermediateStop__c:0;
Double vehicleDestinationToProvider =
m.R1_MIS_NUM_vehicleDestinationToProvider__c != null?
m.R1_MIS_NUM_vehicleDestinationToProvider__c:0;
System.debug('distancias');
System.debug(eventToVehicleDestination);
System.debug(intermediateStopToVehDestin);
System.debug(intermediateStopToProvider);
System.debug(eventToIntermediateStop);
System.debug(vehicleDestinationToProvider);

```

R1_Vision__c EAVision = new R1_Vision__c();

```

if(m.R1_MIS_TEX_MissionCode__c != null)
    EAVision.Name = m.R1_MIS_TEX_MissionCode__c;
if(m.R1_MIS_CHE_ServiceClosed__c != null)
    EAVision.R1_VIS_CHE_ServiceClosed__c = m.R1_MIS_CHE_ServiceClosed__c;
if(m.R1_MIS_PIC_TypeOfVeh__c != null)
    EAVision.R1_VIS_PIC_TypeOfVeh__c = m.R1_MIS_PIC_TypeOfVeh__c;
if(m.R1_MIS_DTI_OcurredAT__c != null)
    EAVision.R1_VIS_DAT_ArrivalDate__c = m.R1_MIS_DTI_OcurredAT__c.Date();
if(m.R1_MIS_PIC_ServiceCodeCore__c != null){
    if(m.R1_MIS_PIC_ServiceCodeCore__c == 'TOW' ||
m.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_TOW'){
        EAVision.R1_VIS_PIC_ServiceType__c = 'Remolcado';
    }else if(m.R1_MIS_PIC_ServiceCodeCore__c == 'ROS' ||
m.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_ROS'){
        EAVision.R1_VIS_PIC_ServiceType__c = 'Reparación in situ';
    }else{
        EAVision.R1_VIS_PIC_ServiceType__c = m.R1_MIS_PIC_ServiceCodeCore__c;
    }
}

```

```

    }
}

if(m.R1_MIS_TEX_WorkshopName__c != null)
    EAVision.R1_VIS_TEX_Workshop__c = m.R1_MIS_TEX_WorkshopName__c;
if(m.R1_MIS_TEX_City_Location__c != null)
    EAVision.R1_VIS_TEX_City__c = m.R1_MIS_TEX_City_Location__c;
if(m.R1_MIS_TEX_State_pickupProvider__c != null)
    EAVision.R1_VIS_TEX_Province__c = m.R1_MIS_TEX_State_pickupProvider__c;
if(m.Id != null)
    EAVision.R1_VIS_MAS_Mission__c = m.Id;
EAVision.RecordTypeId = EAVisionId;
relatedVisionslist.add(EAVision);

///////////////////////////////// Logica para creacion de los conceptos ///////////////////////////////////
R1_ConceptService__c concept = new R1_ConceptService__c();
System.debug('Vemos si es Urbano o Salida');
System.debug(origin);
System.debug(base);
System.debug(midpoint);
System.debug(destiny);
System.debug(conceptPriceMap.keySet());
System.debug('TIPO DE SERVICIO: ' +
mapPAs.get(m.R1_MIS_LOO_Services_Provided__c));
if(mapPAs.get(m.R1_MIS_LOO_Services_Provided__c) != 'TX' &&
mapPAs.get(m.R1_MIS_LOO_Services_Provided__c) != 'GP'){

    // Si hay parada intermedia se tienen que crear varios servicios
    if(midpoint != null){
        // Tenemos que crear dos servicios
        ///////////////////////////////////

        if((origin == base && midpoint == null) || (origin == base && midpoint == null) || (origin
== base && midpoint == origin) || (origin == base && midpoint == origin)){
            System.debug('Porque no entra por aqui');
            List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            concept.Name = 'Urbano';
            concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept.R1_CSE_NUM_Quantity__c = 1;
            concept.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Urbano')).R1_LDT_LOO_Concept_Type__c;
            concept.R2_CSE_TEX_Sort__c = '1';
            relatedConcepts.add(concept);
            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }else{
            System.debug('Porque no entra por aqui 2');
            List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            Double provToEvent = m.R1_MIS_NUM_ProviderToEvent__c != null?
m.R1_MIS_NUM_ProviderToEvent__c:0;
            Double evtToProvider = m.R1_MIS_NUM_EventToProvider__c != null?
m.R1_MIS_NUM_EventToProvider__c:0;
            concept.Name = 'Salida';
            concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;

```

```

        concept.R1_CSE_NUM_Quantity__c = 1;
        concept.R2_CSE_TEX_Sort__c = '1';
        concept.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Salida')).R1_LDT_LOO_Concept_Type__c;
        relatedConcepts.add(concept);
        if((provToEvent+eventToIntermediateStop > 0){
            R1_ConceptService__c conceptKM = new R1_ConceptService__c();
            conceptKM.Name = 'Kilometro';
            conceptKM.R1_CSE_NUM_Quantity__c =
provToEvent+eventToIntermediateStop;
            conceptKM.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            conceptKM.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Kilometro')).R1_LDT_LOO_Concept_Type__c;
            conceptKM.R2_CSE_TEX_Sort__c = '1';
            relatedConcepts.add(conceptKM);
        }

        mapaConceptos.put(EAVision.Name,relatedConcepts);
    }
    //// Creamos el segundo servicio
    if(destiny != null){
        if(((midpoint == null && destiny == null) || (midpoint == null && destiny == midpoint)
|| (midpoint == destiny && destiny == null) || (midpoint == destiny && destiny == midpoint))){
            System.debug('Entra cuando es null el destino ' + destiny);
            // List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            List<R1_ConceptService__c> relatedConcepts =
mapaConceptos.get(EAVision.Name);
            R1_ConceptService__c concept1 = new R1_ConceptService__c();
            concept1.Name = 'Urbano';
            // concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            // concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept1.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept1.R1_CSE_NUM_Quantity__c = 1;
            concept1.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Urbano')).R1_LDT_LOO_Concept_Type__c;
            concept1.R2_CSE_TEX_Sort__c = '2';
            relatedConcepts.add(concept1);
            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }else{
            System.debug('Entra cuando es null el destino ' + destiny);
            // List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            List<R1_ConceptService__c> relatedConcepts =
mapaConceptos.get(EAVision.Name);
            R1_ConceptService__c concept2 = new R1_ConceptService__c();
            Double provToEvent = m.R1_MIS_NUM_ProviderToEvent__c != null?
m.R1_MIS_NUM_ProviderToEvent__c:0;
            Double evtToProvider = m.R1_MIS_NUM_EventToProvider__c != null?
m.R1_MIS_NUM_EventToProvider__c:0;
            concept2.Name = 'Salida';
            // concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            // concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept2.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept2.R1_CSE_NUM_Quantity__c = 1;
            concept2.R2_CSE_TEX_Sort__c = '2';
            concept2.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Salida')).R1_LDT_LOO_Concept_Type__c;
            relatedConcepts.add(concept2);
            if(intermediateStopToVehDestin+vehicleDestinationToProvider > 0){

```



```

        relatedConcepts.add(conceptKM);
    }

    mapaConceptos.put(EAVision.Name,relatedConcepts);
}

}

}else if(mapPAs.get(m.R1_MIS_LOO_Services_Provided__c) == 'GP'){
    System.debug('////////////////////////////////// BUSCAMOS CONCEPTO
PESADAS //////////////////////////////////////////');

    if(midpoint != null){
        if((origin == base && midpoint == null) || (origin == base && midpoint == null ) || (origin
== base && midpoint == origin ) || (origin == base && midpoint == origin )){
            System.debug('Porque no entra por aqui');
            List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            concept.Name = 'Urbano';
            concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept.R1_CSE_NUM_Quantity__c = 1;
            concept.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Urbano'));
            concept.R2_CSE_TEX_Sort__c = '1';
            relatedConcepts.add(concept);
            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }else{
            System.debug('Porque no entra por aqui 2');
            List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            Double provToEvent = m.R1_MIS_NUM_ProviderToEvent__c != null?
m.R1_MIS_NUM_ProviderToEvent__c:0;
            Double evtToProvider = m.R1_MIS_NUM_EventToProvider__c != null?
m.R1_MIS_NUM_EventToProvider__c:0;
            concept.Name = 'Salida';
            concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept.R1_CSE_NUM_Quantity__c = 1;
            concept.R2_CSE_TEX_Sort__c = '1';
            concept.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Salida'));
            relatedConcepts.add(concept);
            if(provToEvent+eventToIntermediateStop > 0){
                R1_ConceptService__c conceptKM = new R1_ConceptService__c();
                conceptKM.Name = 'Kilometro';
                conceptKM.R1_CSE_NUM_Quantity__c =
provToEvent+eventToIntermediateStop;
                conceptKM.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
                conceptKM.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Kilometro'));
                conceptKM.R2_CSE_TEX_Sort__c = '1';
                relatedConcepts.add(conceptKM);
            }

            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }
    }

    //// Creamos el segundo servicio

```



```

        if((midpoint == null && destiny == null) || ( midpoint == null && destiny == midpoint) ||
(midpoint == destiny && destiny == null) || (midpoint == destiny && destiny == midpoint)){
            // List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            List<R1_ConceptService__c> relatedConcepts =
mapaConceptos.get(EAVision.Name);
            R1_ConceptService__c concept1 = new R1_ConceptService__c();
            concept1.Name = 'Urbano';
            // concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            // concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept1.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept1.R1_CSE_NUM_Quantity__c = 1;
            concept1.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Urbano'));
            concept1.R2_CSE_TEX_Sort__c = '2';
            relatedConcepts.add(concept1);
            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }else{
            // List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            List<R1_ConceptService__c> relatedConcepts =
mapaConceptos.get(EAVision.Name);
            R1_ConceptService__c concept2 = new R1_ConceptService__c();
            Double provToEvent = m.R1_MIS_NUM_ProviderToEvent__c != null?
m.R1_MIS_NUM_ProviderToEvent__c:0;
            Double evtToProvider = m.R1_MIS_NUM_EventToProvider__c != null?
m.R1_MIS_NUM_EventToProvider__c:0;
            concept2.Name = 'Salida';
            // concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
            // concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
            concept2.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            concept2.R1_CSE_NUM_Quantity__c = 1;
            concept2.R2_CSE_TEX_Sort__c = '2';
            concept2.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Salida'));
            relatedConcepts.add(concept2);
            if(intermediateStopToVehDestin+vehicleDestinationToProvider > 0){
                R1_ConceptService__c conceptKM = new R1_ConceptService__c();
                conceptKM.Name = 'Kilometro';
                conceptKM.R1_CSE_NUM_Quantity__c =
intermediateStopToVehDestin+vehicleDestinationToProvider;
                conceptKM.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
                conceptKM.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Kilometro'));
                conceptKM.R2_CSE_TEX_Sort__c = '2';
                relatedConcepts.add(conceptKM);
            }

            mapaConceptos.put(EAVision.Name,relatedConcepts);
        }

////////////////////////////////////
    }else{
        if((origin == base && midpoint == null && destiny == null) || (origin == base &&
midpoint == null && destiny == origin) || (origin == base && midpoint == origin && destiny == null)
|| (origin == base && midpoint == origin && destiny == origin)){
            System.debug('ES URBANO');
            List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
            concept.Name = 'Urbano';

```

```

        concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
        concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
        concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
        concept.R1_CSE_NUM_Quantity__c = 1;
        concept.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Urbano'));
        concept.R2_CSE_TEX_Sort__c = '1';
        relatedConcepts.add(concept);
        mapaConceptos.put(EAVision.Name,relatedConcepts);

    }else{
        System.debug('ES SALIDA + KILOMETROS');
        List<R1_ConceptService__c> relatedConcepts = new
List<R1_ConceptService__c>();
        Double provToEvent = m.R1_MIS_NUM_ProviderToEvent__c != null?
m.R1_MIS_NUM_ProviderToEvent__c:0;
        Double evtToProvider = m.R1_MIS_NUM_EventToProvider__c != null?
m.R1_MIS_NUM_EventToProvider__c:0;
        concept.Name = 'Salida';
        concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
        concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
        concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
        concept.R1_CSE_NUM_Quantity__c = 1;
        concept.R2_CSE_TEX_Sort__c = '1';
        concept.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Salida'));
        relatedConcepts.add(concept);
        if((provToEvent+evtToProvider > 0 || (provToEvent + eventToVehicleDestination +
vehicleDestinationToProvider) > 0){
            R1_ConceptService__c conceptKM = new R1_ConceptService__c();
            conceptKM.Name = 'Kilometro';
            conceptKM.R1_CSE_NUM_Quantity__c = provToEvent +
eventToVehicleDestination + vehicleDestinationToProvider + evtToProvider;
            conceptKM.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
            conceptKM.R1_CSE_LOO_Price__c = getGPVehicleType(m,
conceptPriceMapGP.get('Kilometro'));
            conceptKM.R2_CSE_TEX_Sort__c = '1';
            relatedConcepts.add(conceptKM);
        }
        mapaConceptos.put(EAVision.Name,relatedConcepts);
    }
}
}

```

```

System.debug('//////////////////////////////////// //////////////////////////////////////');
/////////);

```

```

    }else if(mapPAs.get(m.R1_MIS_LOO_Services_Provided__c) == 'TX'){
        System.debug('Entra por la parte del Taxi');
        R1_Vision__c ProvVision = new R1_Vision__c();

        if(m.R1_MIS_TEX_MissionCode__c != null)
            ProvVision.Name = 'PROV_' + m.R1_MIS_TEX_MissionCode__c ;
        if(m.R1_MIS_CHE_ServiceClosed__c != null)
            ProvVision.R1_VIS_CHE_ServiceClosed__c = m.R1_MIS_CHE_ServiceClosed__c;
        if(m.R1_MIS_PIC_TypeOfVeh__c != null)
            ProvVision.R1_VIS_PIC_TypeOfVeh__c = m.R1_MIS_PIC_TypeOfVeh__c;
        if(m.R1_MIS_DTI_OcurredAT__c != null)
            ProvVision.R1_VIS_DAT_ArrivalDate__c = m.R1_MIS_DTI_OcurredAT__c.Date();
    }
}

```

```

        if(m.R1_MIS_PIC_ServiceCodeCore__c != null){
            if(m.R1_MIS_PIC_ServiceCodeCore__c == 'TOW' ||
m.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_TOW'){
                ProvVision.R1_VIS_PIC_ServiceType__c = 'Remolcado';
            }else if(m.R1_MIS_PIC_ServiceCodeCore__c == 'ROS' ||
m.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_ROS'){
                ProvVision.R1_VIS_PIC_ServiceType__c = 'Reparación in situ';
            }else{
                ProvVision.R1_VIS_PIC_ServiceType__c = m.R1_MIS_PIC_ServiceCodeCore__c;
            }
        }
    }
    if(m.R1_MIS_TEX_WorkshopName__c != null)
        ProvVision.R1_VIS_TEX_Workshop__c = m.R1_MIS_TEX_WorkshopName__c;
    if(m.R1_MIS_TEX_City_Location__c != null)
        ProvVision.R1_VIS_TEX_City__c = m.R1_MIS_TEX_City_Location__c;
    if(m.R1_MIS_TEX_State_pickupProvider__c != null)
        ProvVision.R1_VIS_TEX_Province__c = m.R1_MIS_TEX_State_pickupProvider__c;
    if(m.Id != null)
        ProvVision.R1_VIS_MAS_Mission__c = m.Id;
    ProvVision.RecordTypeId = RECORDTYPE_PROVIDER_VISION;
    relatedVisionslist.add(ProvVision);

    Decimal totalKM = 0;
    Double providerToPickup = m.R1_MIS_NUM_providerToPickup__c != null?
m.R1_MIS_NUM_providerToPickup__c:0;
    Double dropoffToProvider = m.R1_MIS_NUM_dropoffToProvider__c != null?
m.R1_MIS_NUM_dropoffToProvider__c:0;
    Double pickupToDropoff = m.R1_MIS_NUM_pickupToDropoff__c != null?
m.R1_MIS_NUM_pickupToDropoff__c:0;
    totalKM = providerToPickup + dropoffToProvider + pickupToDropoff;

    if(totalKM <
conceptPriceMap.get(mapKeys.get('Metro')).R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Conc
ept__r.R1_CON_NUM_MinKM__c){
        // Creamos un concepto Urbano
        System.debug('Creamos un concepto Urbano');
        List<R1_ConceptService__c> relatedConcepts = new List<R1_ConceptService__c>();
        List<R1_ConceptService__c> relatedConcepts2 = new
List<R1_ConceptService__c>();
        concept.Name = 'Urbano';
        concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
        concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
        concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
        concept.R1_CSE_NUM_Quantity__c = 1;
        concept.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Urbano')).R1_LDT_LOO_Concept_Type__c;
        concept.R2_CSE_TEX_Sort__c = '1';
        concept.R2_CSE_TEX_DestinationName__c = 'a';
        relatedConcepts.add(concept);
        R1_ConceptService__c conceptDest = new R1_ConceptService__c();
        conceptDest.Name = '-';
        conceptDest.RecordTypeId = RECORDTYPE_DESTINATION_SERVICE_CONCEPT;
        conceptDest.R2_CSE_TEX_Sort__c = '1';
        conceptDest.R2_CSE_TEX_DestinationName__c =
m.R1_MIS_TEX_PlaceName_Location__c;
        conceptDest.R2_CSE_TEX_City__c = m.R1_MIS_TEX_PlaceName_Location__c;
        conceptDest.R2_CSE_DAT_ArrivalDateTime__c = m.R1_MIS_DTI_DropOff__c;
        relatedConcepts.add(conceptDest);
        mapaConceptos.put(ProvVision.Name,relatedConcepts);
    }

```

```

R1_ConceptService__c conceptTAX = new R1_ConceptService__c();
conceptTAX.Name = 'Urbano';
conceptTAX.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
conceptTAX.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
conceptTAX.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
conceptTAX.R1_CSE_NUM_Quantity__c = 1;
conceptTAX.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Urbano')).R1_LDT_LOO_Concept_Type__c;
conceptTAX.R2_CSE_TEX_Sort__c = '1';
conceptTAX.R2_CSE_TEX_DestinationName__c = 'b';

relatedConcepts2.add(conceptTAX);

mapaConceptos.put(EAVision.Name,relatedConcepts2);

}else if(totalKM >=
conceptPriceMap.get(mapKeys.get('Metro')).R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Conc
ept__r.R1_CON_NUM_MinKM__c && totalKM <
conceptPriceMap.get(mapKeys.get('Metro')).R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Conc
ept__r.R1_CON_NUM_MaxKm__c){
    // Creamos un concepto Metropolitano
    System.debug('Creamos un concepto Metropolitano');
    List<R1_ConceptService__c> relatedConcepts = new List<R1_ConceptService__c>();
    List<R1_ConceptService__c> relatedConcepts2 = new
List<R1_ConceptService__c>();
    concept.Name = 'Metropolitano';
    concept.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
    concept.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
    concept.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
    concept.R1_CSE_NUM_Quantity__c = 1;
    concept.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Metro')).R1_LDT_LOO_Concept_Type__c;
    concept.R2_CSE_TEX_Sort__c = '1';
    concept.R2_CSE_TEX_DestinationName__c = 'a';
    relatedConcepts.add(concept);
    R1_ConceptService__c conceptDest = new R1_ConceptService__c();
    conceptDest.Name = '-';
    conceptDest.RecordTypeId = RECORDTYPE_DESTINATION_SERVICE_CONCEPT;
    conceptDest.R2_CSE_TEX_Sort__c = '1';
    conceptDest.R2_CSE_TEX_DestinationName__c =
m.R1_MIS_TEX_PlaceName_Location__c;
    conceptDest.R2_CSE_TEX_City__c = m.R1_MIS_TEX_PlaceName_Location__c;
    conceptDest.R2_CSE_DAT_ArrivalDateTime__c = m.R1_MIS_DTI_DropOff__c;
    relatedConcepts.add(conceptDest);

    mapaConceptos.put(ProvVision.Name,relatedConcepts);
    R1_ConceptService__c conceptTAX = new R1_ConceptService__c();
    conceptTAX.Name = 'Metropolitano';
    conceptTAX.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
    conceptTAX.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
    conceptTAX.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
    conceptTAX.R1_CSE_NUM_Quantity__c = 1;
    conceptTAX.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Metro')).R1_LDT_LOO_Concept_Type__c;
    conceptTAX.R2_CSE_TEX_Sort__c = '1';
    conceptTAX.R2_CSE_TEX_DestinationName__c = 'b';
    relatedConcepts2.add(conceptTAX);
    mapaConceptos.put(EAVision.Name,relatedConcepts2);
}else{
    // Creamos un concepto Kilometros

```

```

        System.debug('Creamos un concepto Kilometros');
        List<R1_ConceptService__c> relatedConcepts = new List<R1_ConceptService__c>();
        List<R1_ConceptService__c> relatedConcepts2 = new
List<R1_ConceptService__c>();
        R1_ConceptService__c conceptKM = new R1_ConceptService__c();
        conceptKM.Name = 'Kilometro';
        conceptKM.R1_CSE_NUM_Quantity__c = totalKM;
        conceptKM.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
        conceptKM.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
        conceptKM.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
        conceptKM.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Kilometro')).R1_LDT_LOO_Concept_Type__c;
        conceptKM.R2_CSE_TEX_Sort__c = '1';
        concept.R2_CSE_TEX_DestinationName__c = 'a';
        relatedConcepts.add(conceptKM);
        R1_ConceptService__c conceptDest = new R1_ConceptService__c();
        conceptDest.Name = '-';
        conceptDest.RecordTypeId = RECORDTYPE_DESTINATION_SERVICE_CONCEPT;
        conceptDest.R2_CSE_TEX_Sort__c = '1';
        conceptDest.R2_CSE_TEX_DestinationName__c =
m.R1_MIS_TEX_PlaceName_Location__c;
        conceptDest.R2_CSE_TEX_City__c = m.R1_MIS_TEX_PlaceName_Location__c;
        conceptDest.R2_CSE_DAT_ArrivalDateTime__c = m.R1_MIS_DTI_DropOff__c;
        relatedConcepts.add(conceptDest);
        mapaConceptos.put(ProvVision.Name,relatedConcepts);
        mapaConceptos.put(ProvVision.Name,relatedConcepts);
        R1_ConceptService__c conceptKM2 = new R1_ConceptService__c();
        conceptKM2.Name = 'Kilometro';
        conceptKM2.R1_CSE_NUM_Quantity__c = totalKM;
        conceptKM2.R2_CSE_NUM_NetAmount__c = m.R1_MIS_NUM_Amount__c;
        conceptKM2.R1_CSE_NUM_GrossCost__c = m.R1_MIS_NUM_Amount__c;
        conceptKM2.RecordTypeId = RECORDTYPE_NORMAL_SERVICE_CONCEPT;
        conceptKM2.R1_CSE_LOO_Price__c =
conceptPriceMap.get(mapKeys.get('Kilometro')).R1_LDT_LOO_Concept_Type__c;
        conceptKM2.R2_CSE_TEX_Sort__c = '1';
        conceptKM2.R2_CSE_TEX_DestinationName__c = 'b';
        relatedConcepts2.add(conceptKM2);
        mapaConceptos.put(EAVision.Name,relatedConcepts2);
    }
}

}

Savepoint sp = Database.setSavepoint();
try{
    System.debug('Aqui me ha llegado para isertar la Vision');
    List<R1_ConceptService__c> lstConFinal = new List<R1_ConceptService__c>();
    System.debug(relatedVisionslist.size());
    insert relatedVisionslist;
    System.debug(mapaConceptos);

    for(Integer i = 0; i<relatedVisionslist.size(); i++) {
        // relatedConcepts[i].R1_CSE_MAS_Vision__c = relatedVisionslist[i].Id;
    }
    System.debug('Las visions');
    System.debug(mapaConceptos.keySet());

    for(R1_Vision__c vis : relatedVisionslist){
        System.debug(vis.Name);
        if(mapaConceptos.containsKey(vis.Name)){

```

```

        List<R1_ConceptService__c> lstCon = mapaConceptos.get(vis.Name);
        System.debug(vis.Id);
        for(R1_ConceptService__c c : lstCon){
            c.R1_CSE_MAS_Vision__c = vis.Id;
            lstConFinal.add(c);
        }
    }
}
insert lstConFinal;
}
catch (Exception e) {
    Database.rollback(sp);
    System.debug('EXCEPTION: ' + e.getStackTraceString());
    CORE_LogErrorHandler.commitExceptionLog(e, 'R1_CLS_MissionTriggerMethods -
insertVisionRecords');
}
}

```

```

public static String getGPVehicleType(R1_Mission__c mission, Map<String, R1_Tariff_Line__c>
mapAUX){
    String aux = '';
    for(String a : mapAUX.keySet()){
        R1_Tariff_Line__c tIAux = mapAUX.get(a);
        if( mission.R1_MIS_PIC_ServiceCodeCore__c == 'ROS' ||
mission.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_ROS'){

if(tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_CHE_ROS__c == true)
{
            aux = tIAux.R1_LDT_LOO_Concept_Type__c;
        }else{
            continue;
        }
    }else
if(tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_PIC_Tipolgy__c ==
mission.R1_MIS_PIC_Type__c){
        if(mission.R1_MIS_PIC_Type__c == 'TRUCK'){
            if(( mission.R1_MIS_PIC_ServiceCodeCore__c == 'ROS' ||
mission.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_ROS') &&
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_CHE_ROS__c == true){
                aux = tIAux.R1_LDT_LOO_Concept_Type__c;
            } else if(mission.R1_MIS_TEX_Platnumber_trailer__c != null &&
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_CHE_Complete__c ==
true){
                aux = tIAux.R1_LDT_LOO_Concept_Type__c;
            }else if(mission.R1_MIS_TEX_Rear_Axles__c != null &&
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_AxesNumber__c !
= null &&
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_AxesNumber__c
== Decimal.valueOf(mission.R1_MIS_TEX_Rear_Axles__c)){
                aux = tIAux.R1_LDT_LOO_Concept_Type__c;
            }else if(mission.R1_MIS_NUM_Weight__c != null &&
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_Minimum_Weight_
__c != null && mission.R1_MIS_NUM_Weight__c >=
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_Minimum_Weight_
__c && mission.R1_MIS_NUM_Weight__c <=
tIAux.R1_LDT_LOO_Concept_Type__r.R1_PRI_LOO_Typology__r.R2_TIP_NUM_Maximum_weight
__c){
                aux = tIAux.R1_LDT_LOO_Concept_Type__c;
            }
        }else if(mission.R1_MIS_PIC_Type__c == 'BUS'){

```

```

        aux = tlAux.R1_LDT_LOO_Concept_Type__c;
    }else if(mission.R1_MIS_PIC_Type__c == 'TRAILER'){
        aux = tlAux.R1_LDT_LOO_Concept_Type__c;
    }
}
}
return aux;
}

```

```

public static void setRequestDateTime (List<R1_Mission__c> misionNews){
    System.debug('SET SERVICE DATE');
    for(R1_Mission__c h : misionNews){
        if(h.R1_MIS_TEX_IdPA__c != null){
            h.R1_MIS_TEX_ServiceTime__c = h.R1_MIS_DTI_PickupTime__c != null ?
h.R1_MIS_DTI_PickupTime__c.time() : null;
            h.R1_MIS_DAT_ServiceDate__c = h.R1_MIS_DTI_PickupTime__c != null ?
h.R1_MIS_DTI_PickupTime__c.date() : null;
            h.R1_MIS_DAT_Request__c = h.R1_MIS_DTI_PickupTime__c != null ?
h.R1_MIS_DTI_PickupTime__c.date() : null;
        }else{
            h.R1_MIS_TEX_ServiceTime__c = h.R1_MIS_DTI_OcurredAT__c != null ?
h.R1_MIS_DTI_OcurredAT__c.time() : null;
            h.R1_MIS_DAT_ServiceDate__c = h.R1_MIS_DTI_OcurredAT__c != null ?
h.R1_MIS_DTI_OcurredAT__c.date() : null;
            h.R1_MIS_DAT_Request__c = h.R1_MIS_DTI_PickupTime__c != null ?
h.R1_MIS_DTI_PickupTime__c.date() : null;
        }
        h.R1_MIS_PIC_Invoice_Status__c = 'Pendiente';
    }
}
}

```

```

public static void setPA (List<R1_Mission__c> misionNews){
    System.debug('RELACION PA');
    Set<String> setPA = new Set<String>();
    for(R1_Mission__c h : misionNews){
        if(h.R1_MIS_PIC_ServiceCodeCore__c == 'TAXI'){
            System.debug(h.R1_MIS_TEX_IdPA__c);
            setPA.add(h.R1_MIS_TEX_IdPA__c);
        }else{
            System.debug('Grua ' + h.R1_MIS_TEX_extId_Provider__c);
            setPA.add(h.R1_MIS_TEX_extId_Provider__c.split('/')[0]);
        }
    }
}

```

```

System.debug(setPA);
List<R1_Services_Provided__c> lstPA = [SELECT Id, R1_SPR_TEX_Code__c
FROM R1_Services_Provided__c WHERE R1_SPR_TEX_Code__c in :setPA];

System.debug(lstPA);
Map<String, R1_Services_Provided__c> mapa = new
Map<String,R1_Services_Provided__c>();

    for(R1_Services_Provided__c v : lstPA){
        mapa.put(v.R1_SPR_TEX_Code__c, v);
    }
    for(R1_Mission__c m : misionNews){
        if(m.R1_MIS_PIC_ServiceCodeCore__c == 'TAXI'){

```

```

        System.debug(mapa);
        if(!Test.isRunningTest()){
            m.R1_MIS_LOO_Services_Provided__c = mapa.get(m.R1_MIS_TEX_IdPA__c).Id;
        }

    }else{
        m.R1_MIS_LOO_Services_Provided__c =
        mapa.get(m.R1_MIS_TEX_extId_Provider__c.split('/')[0]).Id;
    }

    }

}

    public static Id getRecordTypeId(Map<String,Schema.RecordTypeInfo> recordTypeInfo, String
rtDeveloperName){
        return recordTypeInfo.get(rtDeveloperName) != null ?
        recordTypeInfo.get(rtDeveloperName).getRecordTypeId() : null;
    }

    public static void updateVisionRecords (Map<ID, R1_Mission__c> misionNewMap, Map<ID,
R1_Mission__c> misionOldMap){
        Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_VISION =
        Schema.SObjectType.R1_Vision__c.getRecordTypeInfosByDeveloperName();
        Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_SERVICE_CONCEPT =
        Schema.SObjectType.R1_ConceptService__c.getRecordTypeInfosByDeveloperName();

        Id RECORDTYPE_DESTINATION_SERVICE_CONCEPT =
        getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_destinationConcept');
        Id RECORDTYPE_NORMAL_SERVICE_CONCEPT =
        getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_normalConcept');
        Id RECORDTYPE_EA_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
'R1_VIS_RT_EAVision');
        Id RECORDTYPE_PROVIDER_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
'R1_VIS_RT_ProvVision');
        List<R1_Vision__c> lstVisiones = [SELECT Id,R1_VIS_MAS_Mission__c FROM R1_Vision__c
WHERE R1_VIS_MAS_Mission__c in :misionNewMap.keySet() AND RecordTypeId = :
RECORDTYPE_EA_VISION];
        List<R1_ConceptService__c> lstConcepts = [SELECT Id,
R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c FROM R1_ConceptService__c WHERE
R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c in :misionNewMap.keySet() AND
R1_CSE_MAS_Vision__r.RecordTypeId = : RECORDTYPE_EA_VISION];
        Map<ID,R1_Vision__c> mapVisions = new Map<Id,R1_Vision__c>();
        Map<Id, List<R1_ConceptService__c>> mapConcepts = new Map<Id,
List<R1_ConceptService__c>>();
        for(R1_Vision__c a : lstVisiones){
            mapVisions.put(a.R1_VIS_MAS_Mission__c, a);
        }
        for(R1_ConceptService__c c : lstConcepts){
            if(!mapConcepts.containsKey(c.R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c)){
                List<R1_ConceptService__c> lstAux = new List<R1_ConceptService__c>();
                lstAux.add(c);
                mapConcepts.put(c.R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c, lstAux);
            }else{
                List<R1_ConceptService__c> lstAux =
mapConcepts.get(c.R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c);
                lstAux.add(c);
                mapConcepts.put(c.R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c, lstAux);
            }
        }
    }
    System.debug('La vision de EA');

```



```

System.debug(lstVisiones);
List<R1_Vision__c> lstVisionFinal = new List<R1_Vision__c>();
for(Id i : misionNewMap.keySet()){
    R1_Mission__c nueva = misionNewMap.get(i);
    R1_Mission__c vieja = misionOldMap.get(i);
    R1_Vision__c vision = mapVisions.get(i);
    if(nueva.R1_MIS_TEX_MissionCode__c != null && nueva.R1_MIS_TEX_MissionCode__c !=
vieja.R1_MIS_TEX_MissionCode__c){
        vision.Name = nueva.R1_MIS_TEX_MissionCode__c;
    }
    if(nueva.R1_MIS_CHE_ServiceClosed__c != null &&
nueva.R1_MIS_CHE_ServiceClosed__c != vieja.R1_MIS_CHE_ServiceClosed__c){
        vision.R1_VIS_CHE_ServiceClosed__c = nueva.R1_MIS_CHE_ServiceClosed__c;
    }
    if(nueva.R1_MIS_PIC_TypeOfVeh__c != null && nueva.R1_MIS_PIC_TypeOfVeh__c !=
vieja.R1_MIS_PIC_TypeOfVeh__c){
        vision.R1_VIS_PIC_TypeOfVeh__c = nueva.R1_MIS_PIC_TypeOfVeh__c;
    }
    if(nueva.R1_MIS_DTI_OcurredAT__c != null && nueva.R1_MIS_DTI_OcurredAT__c !=
vieja.R1_MIS_DTI_OcurredAT__c){
        vision.R1_VIS_DAT_ArrivalDate__c = Date.valueOf(nueva.R1_MIS_DTI_OcurredAT__c);
    }
    if(nueva.R1_MIS_TEX_WorkshopName__c != null &&
nueva.R1_MIS_TEX_WorkshopName__c != vieja.R1_MIS_TEX_WorkshopName__c){
        vision.R1_VIS_TEX_Workshop__c = nueva.R1_MIS_TEX_WorkshopName__c;
    }
    if(nueva.R1_MIS_TEX_City_Location__c != null && nueva.R1_MIS_TEX_City_Location__c !=
vieja.R1_MIS_TEX_City_Location__c){
        vision.R1_VIS_TEX_City__c = nueva.R1_MIS_TEX_City_Location__c;
    }
    if(nueva.R1_MIS_TEX_State_pickupProvider__c != null &&
nueva.R1_MIS_TEX_State_pickupProvider__c != vieja.R1_MIS_TEX_State_pickupProvider__c){
        vision.R1_VIS_TEX_Province__c = nueva.R1_MIS_TEX_State_pickupProvider__c;
    }

    if(nueva.R1_MIS_PIC_ServiceCodeCore__c != null){
        if(nueva.R1_MIS_PIC_ServiceCodeCore__c == 'TOW' ||
nueva.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_TOW'){
            vision.R1_VIS_PIC_ServiceType__c = 'Remolcado';
        }else if(nueva.R1_MIS_PIC_ServiceCodeCore__c == 'ROS' ||
nueva.R1_MIS_PIC_ServiceCodeCore__c == 'HIGHWAY_ROS'){
            vision.R1_VIS_PIC_ServiceType__c = 'Reparación in situ';
        }else{
            vision.R1_VIS_PIC_ServiceType__c = nueva.R1_MIS_PIC_ServiceCodeCore__c;
        }
    }

    }
    lstVisionFinal.add(vision);

}
update(lstVisionFinal);
}

public static void updatemissionStatus(List<R1_Mission__c> newMissions){
    for(R1_Mission__c m : newMissions){
        if(m.R1_MIS_PIC_PortalStatus__c == 'Reclamado'){
            // m.R2_MIS_PIC_ReviewStatus__c = 'Reclamada';
        }
    }
}
}

```

```

public static void missionStatusController(List<R1_Mission__c> newMissions) {

    //Listas a actualizar
    List<R1_Discrepancy__c> discrepanciesToInsert = new List<R1_Discrepancy__c>();
    List<R1_Discrepancy__c> discrepanciesToDelete = new List<R1_Discrepancy__c>();
    List<R1_Mission__c> missionsToUpdate = new List<R1_Mission__c>();
    List<R1_Vision__c> visionsToUpdate = new List<R1_Vision__c>();

    //Record Types
    Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_VISION =
    Schema.SObjectType.R1_Vision__c.getRecordTypeInfoByDeveloperName();
    Map<String, Schema.RecordTypeInfo> RECORDTYPE_INFO_SERVICE_CONCEPT =
    Schema.SObjectType.R1_ConceptService__c.getRecordTypeInfoByDeveloperName();

    Id RECORDTYPE_DESTINATION_SERVICE_CONCEPT =
    getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_destinationConcept');
    Id RECORDTYPE_NORMAL_SERVICE_CONCEPT =
    getRecordTypeId(RECORDTYPE_INFO_SERVICE_CONCEPT, 'R2_CSE_RT_normalConcept');
    Id RECORDTYPE_EA_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
    'R1_VIS_RT_EAVision');
    Id RECORDTYPE_PROVIDER_VISION = getRecordTypeId(RECORDTYPE_INFO_VISION ,
    'R1_VIS_RT_ProvVision');

    //CUSTOM METADATA CANTIDAD LIMITE
    R2_MDT_limitAmount__mdt limitAmount = getLimitAmount();

    //MAPA DE VISION DE EUA (ID Mision, Vision)
    List<R1_Vision__c> EAVisionList = [SELECT Id, R1_VIS_MAS_Mission__c,
    R1_VIS_ROL_TotalAmount__c, RecordTypeId, R1_VIS_CHE_ServiceClosed__c
    FROM R1_Vision__c
    WHERE R1_VIS_MAS_Mission__c IN :newMissions AND
    RecordTypeId =: RECORDTYPE_EA_VISION];

    Map<String, R1_Vision__c> EAVisionMap = new Map<String, R1_Vision__c>();

    for(R1_Vision__c EAVision : EAVisionList) {
        //si hay mas de una vision solo añadimos la primera
        if (!EAVisionMap.containsKey(EAVision.R1_VIS_MAS_Mission__c)){
            EAVisionMap.put(EAVision.R1_VIS_MAS_Mission__c, EAVision);
        }
    }

    //LISTA DE VISION de prov (ID Mision, Vision)
    List<R1_Vision__c> ProvVisionList = [SELECT Id,
    R1_VIS_MAS_Mission__c, R1_VIS_PIC_TypeOfVeh__c, R1_VIS_PIC_ServiceType__c,
    R1_VIS_DAT_BaseDate__c, R1_VIS_PIC_Damage__c, R1_VIS_DTI_ConcessionaireDate__c,
    R1_VIS_TEX_Workshop__c, R1_VIS_TEX_City__c, R1_VIS_TEX_Province__c,
    R1_VIS_PIC_Subtype__c, R1_VIS_DAT_ArrivalDate__c, R1_VIS_ROL_TotalAmount__c,
    RecordTypeId, R1_VIS_CHE_ServiceClosed__c, R1_VIS_CHE_NoctFest__c
    FROM R1_Vision__c
    WHERE R1_VIS_MAS_Mission__c IN :newMissions AND
    RecordTypeId =: RECORDTYPE_PROVIDER_VISION];

    Map<String, R1_Vision__c> ProvVisionMap = new Map<String, R1_Vision__c>();

    for(R1_Vision__c ProvVision : ProvVisionList) {
        //si hay mas de una vision solo añadimos la primera
        if (!ProvVisionMap.containsKey(ProvVision.R1_VIS_MAS_Mission__c)){
            ProvVisionMap.put(ProvVision.R1_VIS_MAS_Mission__c, ProvVision);
        }
    }
}

```

```
}  
}
```

```
//MAPA DE MISSION (ID Mision, Mision)
```

```
List<R1_Mission__c> missionList = [SELECT Id, R1_MIS_TEX_ACE_OT__c,  
R1_MIS_TEX_MissionCode__c, R1_MIS_PIC_StatusMission__c, R1_MIS_TEX_IdPA__c,  
Name,R1_MIS_CHE_ServiceClosed__c,
```

```
R1_MIS_LOO_Services_Provided__r.R1_SPR_LOO_Platform__r.Name,  
R1_MIS_PIC_PortalStatus__c, R1_MIS_PIC_ReasonRejected__c,  
R1_MIS_PIC_ServiceCodeCore__c, R1_MIS_LOO_Services_Provided__r.R1_SPR_TEX_Code__c,  
R1_MIS_LOO_Services_Provided__r.R1_SPR_LOO_Platform__r.R1_PLA_TEX_Entity__c  
FROM R1_Mission__c  
WHERE Id IN :newMissions];
```

```
Map<Id, R1_Mission__c> missionMap = new Map<Id, R1_Mission__c>(missionList);
```

```
//LISTA DISCREPANCIAS ACTUALES
```

```
List<R1_Discrepancy__c> discrepancyInsertedList = [SELECT Id, R2_DIS_MAS_Mission__c,  
R1_DIS_PIC_Status__c  
FROM R1_Discrepancy__c  
WHERE R2_DIS_MAS_Mission__c IN :newMissions];
```

```
//LISTA CONCEPTOS DE SERVICIO
```

```
List<R1_ConceptService__c> conceptServiceList = [SELECT Id, R1_CSE_MAS_Vision__c,  
R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c, R1_CSE_FOR_ConceptName__c,  
R1_CSE_FOR_Media__c, R1_CSE_FOR_Prestation__c,  
R2_CSE_TEX_Sort__c, R1_CSE_NUM_TotalAmount__c,  
R1_CSE_NUM_Quantity__c, R2_CSE_NUM_NetAmount__c, R1_CSE_LOO_Price__c  
FROM R1_ConceptService__c  
WHERE R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c  
IN :newMissions AND R1_CSE_MAS_Vision__r.RecordTypeId =:  
RECORDTYPE_PROVIDER_VISION  
AND RecordTypeId  
= :RECORDTYPE_NORMAL_SERVICE_CONCEPT  
];
```

```
//LISTA ADICIONALES DE CONCEPTOS SERVICIO
```

```
List<R2_AdditionalConcept__c> additionalConceptList = [SELECT Id,  
R2_ADC_MAS_ServiceConcept__c, R2_ADC_TEX_Name__c  
FROM R2_AdditionalConcept__c  
WHERE  
R2_ADC_MAS_ServiceConcept__r.R1_CSE_MAS_Vision__r.R1_VIS_MAS_Mission__c  
IN :newMissions AND R2_ADC_MAS_ServiceConcept__r.R1_CSE_MAS_Vision__r.RecordTypeId  
=: RECORDTYPE_PROVIDER_VISION];
```

```
//LISTA DE SUPLEMENTOS
```

```
List<R2_Suplement__c> suplemmentsList = [SELECT Id, Name,  
R2_SUP_CUR_SuplementAmount__c, R2_SUP_PIC_SuplementType__c,  
R2_SUP_MAS_Vision__c  
FROM R2_Suplement__c  
WHERE R2_SUP_MAS_Vision__r.R1_VIS_MAS_Mission__c  
IN :newMissions AND R2_SUP_MAS_Vision__r.RecordTypeId =:  
RECORDTYPE_PROVIDER_VISION];
```

```
for (R1_Mission__c mission : newMissions) {
```

```

R1_Vision__c ProvVision = ProvVisionMap.get(mission.id);
R1_Vision__c EAVision = EAVisionMap.get(mission.id);

if (ProvVision != null && EAVision != null) {
    Double diffAmount = ProvVision.R1_VIS_ROL_TotalAmount__c -
EAVision.R1_VIS_ROL_TotalAmount__c;

    System.debug('--- VISION PROVEEDOR AMOUNT = ' +
ProvVision.R1_VIS_ROL_TotalAmount__c);
    System.debug('--- VISION EUA AMOUNT = ' + EAVision.R1_VIS_ROL_TotalAmount__c);
    System.debug('--- DIFERENCIA = ' + diffAmount);

    if (!ProvVision.R1_VIS_CHE_NoctFest__c) {

        System.debug('--- ORIGEN: PORTAL ---');

        if (mission.R1_MIS_PIC_PortalStatus__c == 'Aceptado') {
            continue;
        }
        else{

            System.debug('--- LOGICA DISCREPANCIAS ---');
            if (diffAmount > limitAmount.R1_MTD_NUM_limit__c) {

                if (mission.R1_MIS_PIC_PortalStatus__c == 'Pendiente') {

                    System.debug('--- ESTADO PENDIENTE ---');
                    //habria que desmarcar el check

                }
                else if (mission.R1_MIS_PIC_PortalStatus__c == 'Reclamado') {

                    System.debug('--- ESTADO RECLAMADO ---');

                    Map<String, List<R1_Discrepancy__c>> resultMap =
handleReclaimedStatus(mission, ProvVision, EAVision, discrepanciesToDelete,
discrepanciesToInsert, discrepancyInsertedList, diffAmount);
                    discrepanciesToDelete = resultMap.get('toDelete');
                    discrepanciesToInsert = resultMap.get('toInsert');

                }
                else if (mission.R1_MIS_PIC_PortalStatus__c == 'Finalizado'){

                    System.debug('--- ESTADO FINALIZADO ---');
                    System.debug('--- SE ABRE EL POP UP ---');

                    R1_Mission__c updatedMission = new R1_Mission__c(Id = mission.Id,
R1_MIS_PIC_PortalStatus__c = 'Pendiente confirmar');
                    missionsToUpdate.add(updatedMission);
                }

            } else {

                if (mission.R1_MIS_PIC_PortalStatus__c == 'Finalizado'){

                    System.debug('--- ESTADO FINALIZADO ---');
                    System.debug('Mission: ' +
missionMap.get(mission.Id).R1_MIS_LOO_Services_Provided__r.R1_SPR_TEX_Code__c);

```

```

        System.debug('Mission: ' + missionMap.get(mission.Id));
        handleFinishedStatus(missionMap.get(mission.Id), ProvVision,
conceptServiceList, supllementsList, additionalConceptList);
    }
}
}
}
else{

    System.debug('--- ORIGEN: PLATAFORMAS ---');

}

} else {
    System.debug('Error: ProvVision o EAVision es nulo para la misi3n con ID: ' +
mission.Id);
}
}

    updateDiscrepancies(discrepanciesToDelete, discrepanciesToInsert, missionsToUpdate);
}

private static R2_MDT_limitAmount__mdt getLimitAmount() {
    R2_MDT_limitAmount__mdt limitAmount;

    try {
        limitAmount = [SELECT Id, R1_MTD_NUM_limit__c
                        FROM R2_MDT_limitAmount__mdt
                        LIMIT 1];
    } catch (QueryException e) {
        System.debug('Error al recuperar R2_MDT_limitAmount__mdt: ' + e.getMessage());
    }

    return limitAmount;
}

private static void updateDiscrepancies(List<R1_Discrepancy__c> discrepanciesToDelete,
List<R1_Discrepancy__c> discrepanciesToInsert, List<R1_Mission__c> missionsToUpdate) {
    try {
        System.debug('--- ACTUALIZAMOS LOS DATOS ---');

        if (!discrepanciesToDelete.isEmpty()) {
            delete discrepanciesToDelete;
        }

        if (!discrepanciesToInsert.isEmpty()) {
            insert discrepanciesToInsert;
        }

        if (!missionsToUpdate.isEmpty()) {
            update missionsToUpdate;
        }
    } catch (DmlException e) {
        System.debug('Error al realizar operaciones DML: ' + e.getMessage());
    }
}

private static Map<String, List<R1_Discrepancy__c>> handleReclaimedStatus(R1_Mission__c
mission, R1_Vision__c ProvVision, R1_Vision__c EAVision, List<R1_Discrepancy__c>

```

```

discrepanciesToDelete, List<R1_Discrepancy__c> discrepanciesToInsert,
List<R1_Discrepancy__c> discrepancyInsertedList, Decimal diffAmount){

    System.debug('--- ELIMINAMOS LAS DISCREPANCIAS RELACIONADAS A ESA MISION
---');

    for(R1_Discrepancy__c discrepancy : discrepancyInsertedList){
        if (discrepancy.R2_DIS_MAS_Mission__c == ProvVision.R1_VIS_MAS_Mission__c) {
            discrepanciesToDelete.add(discrepancy);
        }
    }

    System.debug('--- CREAMOS LA DISCREPANCIA NUEVA ---');

    R1_Discrepancy__c auxDiscrepancy = new R1_Discrepancy__c();

    auxDiscrepancy.Name = mission.Name + '-DIS';
    auxDiscrepancy.R2_DIS_MAS_Mission__c = ProvVision.R1_VIS_MAS_Mission__c;
    auxDiscrepancy.R1_DIS_DAT_CreationDate__c = date.today();
    auxDiscrepancy.R1_DIS_DAT_StartDate__c = date.today();
    auxDiscrepancy.R1_DIS_PIC_Status__c = 'Pendiente';
    auxDiscrepancy.R2_DIS_CUR_DiffAmount__c = diffAmount;
    auxDiscrepancy.R2_DIS_CUR_EaAmount__c = EAVision.R1_VIS_ROL_TotalAmount__c;
    auxDiscrepancy.R2_DIS_CUR_ProviderAmount__c =
ProvVision.R1_VIS_ROL_TotalAmount__c;
    auxDiscrepancy.R2_DIS_TEX_Product__c = mission.R1_MIS_TEX_Product_Code__c;

    discrepanciesToInsert.add(auxDiscrepancy);

    Map<String, List<R1_Discrepancy__c>> resultMap = new Map<String,
List<R1_Discrepancy__c>>();
    resultMap.put('toDelete', discrepanciesToDelete);
    resultMap.put('toInsert', discrepanciesToInsert);

    return resultMap;
}

private static void handleFinishedStatus(R1_Mission__c mission, R1_Vision__c ProvVision,
List<R1_ConceptService__c> conceptServiceList, List<R2_Suplement__c> suplemmentsList,
List<R2_AdditionalConcept__c> additionalConceptList) {

    System.debug('--- SE LLAMA A FINALIZACION DE SERVICIO ---');

    try {
        List<R1_ConceptService__c> conceptServiceListAux = new
List<R1_ConceptService__c>();
        List<R2_Suplement__c> suplemmentsListAux = new List<R2_Suplement__c>();

        //Obtenemos los conceptos y suplementos asignados a la vision
        for (R1_ConceptService__c conceptService : conceptServiceList) {
            if (conceptService.R1_CSE_MAS_Vision__c == ProvVision.Id) {
                conceptServiceListAux.add(conceptService);
            }
        }

        for (R2_Suplement__c suplemment : suplemmentsList) {
            if (suplemment.R2_SUP_MAS_Vision__c == ProvVision.Id) {
                suplemmentsListAux.add(suplemment);
            }
        }
    }
}

```

```

        if(!Test.isRunningTest()){
            if(!conceptServiceListAux.isEmpty()){
                System.debug('Mission: ' +
mission.R1_MIS_LOO_Services_Provided__r.R1_SPR_TEX_Code__c);
                R2_CLS_EndOfServiceRequest request =
R2_CLS_EndOfServiceRequest.getEndOfServiceRequest(mission, ProvVision,
conceptServiceListAux, additionalConceptList, suplemmentsListAux, null);
                String JSONData = JSON.serialize(request, true);
                System.debug(JSONData);

                R2_CLS_SendEndofService.sendEndofServiceFuture(JSONData, mission.Id,
ProvVision.Id, null);
            }
        }
    } catch (Exception e) {
        System.debug('Error en handleFinishedStatus: ' + e.getMessage());
    }
}
}

```