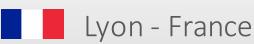




# iDempiere PAYROLL PLUGIN

iDempiere World Conference 2019



Luis Amesty

luisamesty@gmail.com



# iDempiere PAYROLL PLUGIN







- Author: Luis Amesty Linares
- Email: <u>luisamesty@gmail.com</u>
- Linked-in: <u>www.linkedin.com/in/luisamesty</u>
- Idempiere Group:
- https://wiki.idempiere.org/en/User:Luisamesty
- Profession: Computer Nomad, living in Spain





## WIKI PUBLICATIONS

#### **Extended Location**

Extension on demographics aspects on Idempiere.

https://wiki.idempiere.org/en/Plugin: Extended Location

### Themes Amerpsoft:

Collection of themes from different contributors. Idempiere concepts on theme handling. https://wiki.idempiere.org/en/Plugin: Themes Amerpsoft

### Financial Plugin:

Processes for multiple currency, some basic jaspersoft reports and closing journal entry. https://bitbucket.org/amerpsoft/amerpsoft-idempiere-community/src/default/org.amerpsoft.com.idempiere.financial/

#### Building iDempiere plugins with Maven:

A tutorial for building plugin's jar file under maven for version 6.2

https://wiki.idempiere.org/en/Building iDempiere Plugins with Maven

iDempiere World Conference 2019



## PRIVATE PLUGINS

### Private Plugins

- ✓ Business Partner processes
- ✓ Document reactivate tools
- ✓ Material processes and average cost
- ✓ Withholding based on LCO from GlobalQss
- ✓ Financial and accounting with dual currency
- ✓ Cash and banks, Special Forms
- ✓ This particular object of the presentation,
- Payroll plugin





## PAYROLL PLUGIN - INTRODUCTION

Venezuelan Labor Law (developed for Tamanaco)



### Requirements:

- No Workers: 250
- Contracts: Weekly, Monthly, Biweekly, Directors and Sales Reps
- Attendance: On several factories for week contracts.
- Locations: 6
- Production Cost Accounting
- User administrated
- User preparing Roles
- Confidential Information





## PAYROLL PLUGIN - INTRODUCTION

- Libero Payroll and LVE plugin Analysis
  - Improve Old System without changing completely (As user Point of view)
  - Lack of knowledge of Open Source policies
  - Lack of knowledge of Idempiere Script Engine
  - Lack of knowledge of Libero business logic
  - Uncertainty in aspects such as:
    - Role Access control for confidential information
    - Default Quantity values
    - Period reopening
    - Document Accounting process
    - Document Reactivating process
- Face a Challenge to make a new plugin from Scratch





## PAYROLL PLUGIN - INTRODUCTION

### Considerations

### **Document Oriented**

Each Payroll receipt will generate its own accounting fact lines.

### Payroll roles

- ✓ Menus
- ✓ Payroll Processes Restriction
- ✓ Payroll Report Limitations
- ✓ Data on Windows





- 1. Payroll Process
- 2. Contract Types
- 3. Attendance
- 4. Receipts Generation
- 5. Payroll Periods
- 6. Multiple receipts on periods
- 7. Posting and reactivation
- 8. Payment Concepts

- 9. Default Values
- 10. Calculation Order
- 11. Public variables
- 12. Rules
- 13. Workers as Business Partners
- 14. Reports on Jaspersoft
- 15. Work Force
- 16. Organization Chart





### Payroll Process:

Set of <u>accounting documents</u> and/or <u>payments</u> that has to be done with group of workers from a particular organization.

### Different payroll processes can be handled:

- NN Payroll Payment
- NV Vacations
- NU End of year bonus
- TI Food Ticket
- NP Law Social Benefits provision
- TO Other Social Benefits
- PO Ocasional Loans
- PL Law Social Benefit Final Payment
- PI Law Interest payment on Social Benefit Balance





### Contract Types:

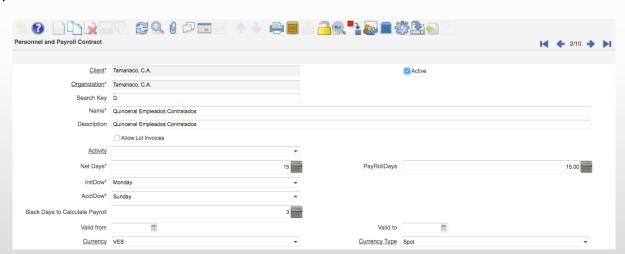
Flexibility to handle several types of Contracts, with possibility of payments with different frequency schemes.

### Frequencies:

- Weekly
- Biweekly
- Monthly
- Custom

### Other Attributes.:

- Slack Days
- Dow Initial
- Dow Account







### Attendance:

To perform the calculation of the payroll process based on the attendance of the worker, through a model of shifts and workdays. Mass assistance loading procedures from clocks are applied for this process.





Generation of receipts for any payroll process:

The generation of receipts is done through a procedure of predefined concept templates in each period, reducing the work associated only to occasional and / or variable concepts.





### Payroll Periods:

- The payroll periods are predefined in conjunction with the fiscal periods of the ERP Client.
- Periods are defined for each Payroll Process and Contract
- Admin user open and close periods
- Admin User may reopen periods and adjust historical receipts.





### Multiple Receipts:

Several receipts from the same process and contract can be registered in the same period, in order to make adjustments to errors or omissions and that are accounted for in the same period.





Posting and Reactivation of Receipts:

Posting (Completing) of receipts by Batch Periods or individually. Reactivation allowed to carry out accounting adjustments, individually by receipts or in batches.





### Payment Concepts:

Concepts based on Formulas Script (JavaScript) with an own calculation engine that use rules, predefined public variables and others dependent on the same concepts, as well as historical values.

They can be User customizable.





### **Default Values:**

Each concept has associated a formula script that establishes a default value for Quantity(CN).

(Quantity – CN Variable of the formula), which facilitates the creation of templates.





### Calculation Order (CalcOrder)

Concept attribute used to perform calculation order in receipt and presentation. Sequence of calculation.

DIFICIO TAMAN	NACO, ARAL	JRE, PORTUGI	JESA	ontable: 26/08/18
DE NÓMINA	DE TRAB	AJADORES	(del Periodo: 20/	08/18 al 26/08/18) No.130191
CHE JIMENEZ				
	Cargo: Co	OSTURERO		
52,90			idad: ACARIGUA	
Cantidad			Asignaciones	Deducciones
52	2.90			ncent
(	0,00	0,00		3.1.50pt
(	0,00	0,00		
32	2,00	7,06	7,06	
90	0,00	90,00	90,00	
4	1,00	1,44	1,44	
4	1,00	2,40	2,40	
(	0,00	0,00		
2	2,00	50,45	50,45	
1	1,00	25,23	25,23	
4	1,00	0,49		0,49
(	),50	0,07		0,0
1	1,00	1,77		1,7
_ (	0,50	0,89		0,8
			176,58	3,22
Qua	antity		Total (BsS):	173,36
_				
	Firma			Huella
	DE NÓMINA CHE JIMENEZ N 23 52,90 Cantidad 55,60 6,60 6,60 6,60 6,60 6,60 6,60 6,6	DE NÓMINA DE TRAB. CHE JIMENEZ N 23 Cargo: Ct 52,90 Id.: V12096 Cantidad 52,90 0,00 0,00 0,00 32,00 90,00 4,00 4,00 0,00 2,00 1,00 4,00 0,50 1,00 0,50 Cantity	DE NÓMINA DE TRABAJADORES CHE JIMENEZ N 23 Cargo: COSTURERO 52,90 Id.: V120905526 Local Cantidad  52,90 52,90 0,00 0,00 0,00 0,00 32,00 7,06 90,00 90,00 4,00 1,44 4,00 2,40 0,00 0,00 2,00 50,45 1,00 25,23 4,00 0,49 0,50 0,07 1,00 1,77 0,50 0,89  Quantity	DENÓMINA DE TRABAJADORES  CHE JIMENEZ  N 23





### PUBLIC VARIABLES (1):

Each Payment Concept or Default Value has an associated a formula script that establishes a calculation rule to generate a result value.

This result value is based on Quantity, Script formula and Pre-Defined Variables

### Pre-Defined Variables types:

- ✓ Public Variables depending on period Dates
- ✓ Array Variables from Previous calculated payment concept result
- ✓ Employee related attributes
- ✓ Process and Contract attributes
- ✓ Rules
- ✓ Last Period values





#### Standard Variables:

R\_ASIG, R\_DEDUC, R\_TOTAL

RA\_ASIG, RA\_DEDUC, RA\_TOTAL, RE\_ASIG, RE\_DEDUC, RE\_TOTAL, WorkingDaysDT DIAS, HORAS, PNRM, DT, DTREC, DTPER, DTOK, NONLABORDAYS, LABORDAYS, HOLLIDAYS BUSINESSDAYS, NONBUSINESSDAYS, WEEKENDDAYS, NONWEEKENDDAYS; UNIDA DTRIBUTARIA.SBMIN

QTY\_HND,QTY\_HNN,QTY\_HED,QTY\_HEN RSU HND,RSU HNN,RSU HED,RSU HEN

IS FAOV.IS SALARIO.IS INCES.IS SSO.IS ARC. IS SPF. IS DESCANSO

IS\_FERIADO,IS\_UTILIDAD,IS\_PRESTACION, IS\_VACACION

 $R\_FAOV, R\_SALARIO, R\_INCES, R\_SSO, R\_ARC, R\_SP$ 

R\_DESCANSO,R\_FERIADO,R\_UTILIDAD,R\_VACACION, R\_PRESTACION,

AM Contract, AM Process, AM Shift, AM Payroll Mode, AM Status,

AM\_IncomeDate, AM\_PaymentType, AM\_CivilStatus, AM\_Sex, AM\_Spouse

AM IsPensioned, AM IsStudyng, AM IsMedicated, AM BirthDate, AM Currency,

REC InitDate, REC EndDate, ACCT Date, REF InitDate, REF EndDate

AM\_Workforce, AM\_Department, AM\_Location, AM\_Project, AM\_Activity

AM Jobtitle, AM Jobstation, AM Jobunit

#### Array Variables:

Quantity Value in Payroll Line:

QT XXXXXXX where XXXXXXX Concept Type TAG

Result Calculated Value in Payroll Line:

RS XXXXXXX where XXXXXXX Concept Type TAG

#### Process and Contract Variables:

(For IF or CASE Sentences)

AM\_Process: Process Associated with Receipt AM\_Contract: Contract Associated with Receipt

#### Standard Default Variables:

DV\_SALARY: Employee Monthly Salary, Calculated on Average last 6 months

DV HOURS: Payroll Period in HOURS Efective (Labor DAYS \* 8)

DV\_DAYS: Payroll receipt Period Days

 ${\tt DV\_TRANSPORTBONUS:} Transport\ Bonus\ Calculated\ depending\ on\ Attendance\ days.\ Returns$ 

Working Days from the Payroll Period

DV\_ATTENDANCEBONUS: Attendance Bonus Calculated depending on Attendance days.

Returns Working Days from the Payroll Period.

 ${\tt DV\_RESTDAYS:} \ {\tt Rest Days Calculated depending on Attendance days.} \ {\tt Rest Days Calculated depending days.} \ {\tt Rest Days Calculated depending days.} \ {$ 

from the Payroll Period

DV\_HOLLIDAYS: Hollidays Calculated depending on Attendance days. Return Non Working

Days from the Payroll Period.

DV\_SALVAC: Return Accrued Salary during last period in term of Monthly Salary.

Summarizing all Concepts Types where IS\_VACAION = 'Y'. Returns Salary on Employee on AMN Employee table if not exists historic payments.

#### Employee Public Variables: (For IF or CASE Sentences)

AM\_Shift: Employee's Shift Value.

AM\_PayrollMode: Employee Payroll Mode (AMN\_payrollmode Reference: S:Stándar,

A:Attendance, H:Hours, B:Both)

AM Status: Employee's status (AMN Status Reference: A:Active, R:Retired, S:Suspended,

V:Vacation.)

AM\_IncomeDate: Employee's IncomeDate.

AM PaymentType: Employee's payment Type (AMN paymenty pe Reference: C:Check,

D:deposit, C:Cash, O:Other)

AM CivilStatus: Employee's Civil status (AMN civilstatus D:Divorced, M:Married, O:Other,

S:Singled, W:Widowed)

AM\_Sex: Employee's sex (M:Male, F:Female) AM\_Spouse: Employee's marital status (Y:Yes, N:No) AM\_IsPensioned: Employee's Persion (Y:Yes, N:No) AM\_IsStudy ng: Employee's Student Status (Y:Yes, N:No) AM IsMedicated: Employee's Medication status (Y:Yes, N:No)

AM\_BirthDate: Employee's BirthDay.

19

iDempiere World Conference 2019



#### LAST VALUE:

Last Value variables materialized as QT LASTVALUE or RS LASTVALUE.

They can be used as variables of given concept.

They will return the value Entered or Calculated in the last period before current period time end.

QT\_LASTVALUE or LASTVALUE: They will return the Quantity.

RS LASTVALUE: It will return the amount calculated.

#### Comment:

This feature was recently added, because users ask me to add fields on on employee table, constantly. So I decided to carry on this, making a "Reference Concept" that keeps this value from the last receipt ordered by end date. This way I can have as many values as I need for the employees, without affecting employee table and business logic.





Rules: Rules are special entities defined on Payroll Plugin, similar to Idempiere Rules. Using a special table AMN\_Rules, user may define custom code, to calculate a value, based on predetermined Predefined Parameters:

AMN\_Payroll\_ID, AMN\_Employee\_ID, InvDateIni, InvDateEnd, InvDateAcc, RefDateIni, RefDateEnd, EmployeeIncomeDate, EmployeeNYears, Record\_ID, AD\_Client\_ID, AD\_Org\_ID, AD\_User\_ID, AMN\_Shift\_ID.

Return Value: RV\_XXXXXXXX. (Where XXXXX is the Rule Tag)

#### Sample of Fractioned End of Year Bonus:



6



### Rule Sample

<u>Client</u> *	Maracaibo Country Club	<u>Organization</u>	•
Rule Order	100,030		
Search Key	beanshell:RV_FDAYS_THISYEAR		☑ Active
Name*	RV_FDAYS_THISYEAR		
Description	RV_FDAYS_THISYEAR Días transcurridos desde el 01/01/XXXX		
Comment/Help	Dias transcurridos desde el 01/01/XXXX Util para calcular valores fraccionados como VACACIONES, UTILIDADES y otros	beneficios	
Data Access Level	Client+Organization		
Entity Type*	AMERP Localization		
Event Type*	Process		
Rule Type*	JSR 223 Scripting APIs		
Script	// RV. FDAYS. THISYEAR import org.compiere.model.*; import org.compiere.model.*; import java.math.*; import java.math.*; import java.sql.Timestamp; import java.sql.Timestamp; import java.sql.Timestamp; import java.sql.Timestamp; import java.util.ArrayList; import java.util.ArrayList; import java.util.GregorianCalendar; import java.util.Properties; import java.ut		
		iDa	mniere World Conference 2019



### Workers:

They are managed according to the criteria of Business Partners in the Idempiere ERP. Additional Tables with the other attributes necessary for the development of the module are added.

#### Additional Tables:

- Employee Attributes Table
- Salary Historic
- Salary Tax
- Salary Social Benefits and Taxes





### Work Force

### Means:

- Sales
- Administrative
- Direct Labor
- Indirect Labor
- Management

They are associated to the workers based on Job Title. Basic for segregating Production Cost Accounting.





### Organizational Chart

### Based on:

- Job Units
- Job Stations
- Job Titles

U111001 E11100105	Armado de Piezas Guantes Corte cuerdas y desbaste de cuerdas	2	
	C11100108-Cortador de Accesorios		MOD-Mano de Obra Directa
	C11100105-Cortador de Cuerdas		MOD-Mano de Obra Directa
E11100106	Corte Vivo y Corte de Ribete	1	
	C11100106-Almacenista (Materia Prima) ALV152-JOSE GREGORIO ALVARADO MACIAS GIM159-MANUEL ALBERTO GIMENEZ TORRES		MOD-Mano de Obra Directa
E11100107	Desbastado de Piel (ribete y piezas)	1	
	C11100107-Ayudante de Estantería		MOD-Mano de Obra Directa
	C11100107-Desbastador de Piel ESC020-ALEXANDER RAFAEL ESCALONA JEREZ		MOD-Mano de Obra Directa
E11100110	Estampado	1	
	C11100110-Estampador BET071-JOSE ALFREDO BETANCOURT PEÑA BUSB41-FRANKLIN ENRIQUE BUSTILLO ESCOBAR		MOD-Mano de Obra Directa
E11100102	Troquel forro piel	2	
	C11100102-Almacenista (Materia Prima) MEL070-JAIME ENRIQUE MELENDEZ OVIEDO		MOD-Mano de Obra Directa

To be used on manufacturing costs calculations combined with Work Force





# Calendar Calendar integration with C\_Calendar Table.

Active	Name	Date	Country
<b>V</b>	Año nuevo	01/01/2020	Y_Y_Venezuela
<b>√</b>	Noche buena	12/31/2019	Y_Y_Venezuela
<b>√</b>	Navidad 25	12/25/2019	Y_Y_Venezuela
✓	Navidad 24	12/24/2019	Y_Y_Venezuela
✓	Dia de la Raza	10/12/2019	Y_Y_Venezuela
√	Natalicio Libertador	07/24/2019	Y_Y_Venezuela
✓	Firma de Acta de Independencia	07/05/2019	Y_Y_Venezuela
✓	Batalla de Carabobo	06/24/2019	Y_Y_Venezuela
✓	1° de Mayo	05/01/2019	Y_Y_Venezuela
✓	Viernes Santo	04/19/2019	Y_Y_Venezuela
✓	Declaración de Independencia	04/19/2019	Y_Y_Venezuela

6



## PAYROLL WINDOW

### Payroll Window

The basic idea was to develop a window where the user worked as if he were facing a spreadsheet by preparing a receipt manually with a heading and lines. Similar to a Sales Invoice Documents.

<u>Heading</u>: Worker identification, Process, Job Title, Initial and End Period Dates, Account Date, Currency, Document No and others.

<u>Lines</u>: Calculation Order, Description, Quantity, Amount Allocated, Amount Deducted, Amount Calculated.

Select: Process, Contract and Period

Create Header and Lines with a Smart Process:

- ☐ Individual Employee
- ☐ All employees based on a group criteria (Process –Contract Period)
- \* IMPORTANT: Consider Role-Access for Process, Contract and Period.





## PAYROLL WINDOW

### Payroll Window. (To meet this goal)

Special Attributes Added to Concepts:

OptMode: Reference - Balance - Allocation(+) - Deduction(-) - Provision - Withholding.

Sign: Debit-Credit.

Rule: Fixed - Variable. (Refers to concept condition Fixed or Variable)

isRepeat: Non Repeatable – Repeatable on receipt.

isShow: YES-NO

Formula(Script): We already talked about

Default Logic(Script default Value): We already talked about

CalcOrder: Order for presenting concept on receipt and calculation.





## PAYROLL WINDOW



Talk is cheap, Show me the code. Linus Torvalds.





### Payroll Windows: (Work ON)

Client: Klismo

Payroll Processes (NN-PL)

Payroll Contracts (ME-SE)

Payroll Role Access

Payroll Concept Types for NN

Payroll Periods for NN

Payroll Preparing NN for ME and SE

Reports for NN

Payroll Concept Types and Rules for PL

Payroll Preparing Social Benefits PL

Receipts for PL





### **Payroll Processes**



Personnel and Payroll Proc	ess	
<u>Client</u> *	Klismo Diseños	
Organization*	Klismo Diseños C.A.	
Search Key	PL	
Name*	Social Benefits Liquidation	
Description	Liquidación de Prestaciones Sociales	
Process Value*	PL	
	Allow Lot Invoices	



## **Payroll Contract**

Personnel and Payroll Cont	ract			< 2/2 > >	
<u>Client</u> *	Klismo Diseños		Active		
Organization*	Klismo Diseños C.A.				
Search Key	SE				
Name*	Weekly				
Description	Weekly Payment				
	Allow Lot Invoices				
Activity	<u></u>				
Net Days*	7 📓	PayRollDays		7.00	
InitDow*	Monday				
AcctDow*	Friday				
Slack Days to Calculate	0 🖼				
Payroll Valid from		Valid to	<b>m</b>		
Currency				-	
Currency	VEG	<u>Currency Type</u>	Орог		
			iDemn	iere World Conferenc	2019
			ірепір	icie vvoria comercia	2015



## **Payroll Role Access**

Perso	nnel and Payroll Role Ac	cess				I< <
Sorted: #15						
	Payroll Process	Payroll Contract	Active	Role	Process Value	Name
	NN-Payroll NN	ME-Monthly	<b>√</b>	Klismo Diseños Admin	NN	Klismo Diseños Admin - NN - ME
5	NN-Payroll NN	ME-Monthly	✓	System Administrator	NN	System Administrator - NN - ME
	NN-Payroll NN	ME-Monthly	✓	Klismo Diseños RRHH	NN	Klismo Diseños RRHH - NN - ME
	NN-Payroll NN	SE-Weekly	✓	Klismo Diseños Admin	NN	Row 2 Diseños Admin - NN - SE
	NO-Other Payments a	ME-Monthly	✓	Klismo Diseños Admin	NO	Klismo Diseños Admin - NO - ME
	NP-Social Benefits	ME-Monthly	✓	Klismo Diseños Admin	NP	Klismo Diseños Admin - NP - ME
	NU-Nómina de Utilida	ME-Monthly	<b>√</b>	Klismo Diseños Admin	NU	Klismo Diseños Admin - NU - ME
	NV-Nómina de Vacaci	ME-Monthly	✓	Klismo Diseños Admin	NV	Klismo Diseños Admin - NV - ME
	PI-Nómina de Interese	ME-Monthly	✓	Klismo Diseños Admin	PI	Klismo Diseños Admin - PI - ME
	PL-Social Benefits Liq	ME-Monthly	✓	Klismo Diseños Admin	PL	Klismo Diseños Admin - PL - ME
	PL-Social Benefits Liq	SE-Weekly	<b>√</b>	Klismo Diseños Admin	PL	Klismo Diseños Admin - PL - SE
	PO-Nómina de Présta	ME-Monthly	✓	Klismo Diseños Admin	PO	Klismo Diseños Admin - PO - ME
	PR-Nómina de Anticip	ME-Monthly	✓	Klismo Diseños Admin	PR	Klismo Diseños Admin - PR - ME
	TI-Nómina de Benefici	ME-Monthly	✓	Klismo Diseños Admin	TI	Klismo Diseños Admin - TI - ME
	TO-Nómina Otros Ben	ME-Monthly	✓	Klismo Diseños Admin	то	Klismo Diseños Admin - TO - ME
						iDempiere World Conference 20





### Payroll Concept Types for NN

Simple Payroll: Basic Salary multiplied by working days Bonus: 20 % of Salary if 90 % Attendance
15 % of Salary if 50% to 89 % Attendance
0 % if less than 50%

```
100.050
SB: Basic Salary
Reference - Fixed
Formula: CN
Default Value:
if (SB > SBMIN) {
SB + 0;
} else {
SBMIN +0;
}
```

```
100.100
DIAS: Working Days
Reference - Fixed
Formula: CN
Default Value:
var Contrato= AM_Contract;
switch(Contrato) {
   case 'ME': 30;
   break;
   case 'QU': 15;
   break;
   case 'SE': 7;
   break;
   default: 30;
}
```

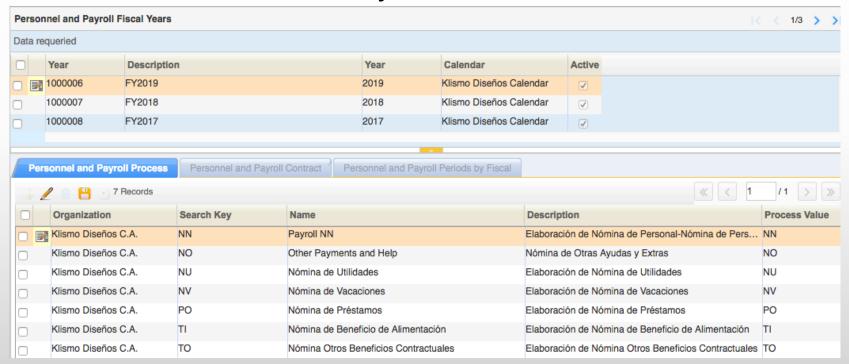
```
100.120
SALARIO: Salary
Allocation - Fixed
Formula:
(QT_SB/30)*QT_DIAS*CN
Default Value:
```

```
100.125
BONUS: BONUX
Allocation - Fixed
Formula:
var P_Attend = (QT_DIAS / DTPER) * 100;
if (P_Attend >= 90) {
   RS_SB * 0.2;
} else if (P_Attend >= 50) {
   RS_SB * 0.15;
} else {
   0;
}
Default Value: 1
```

iDempiere World Conference 2019



### Payroll Periods (1)

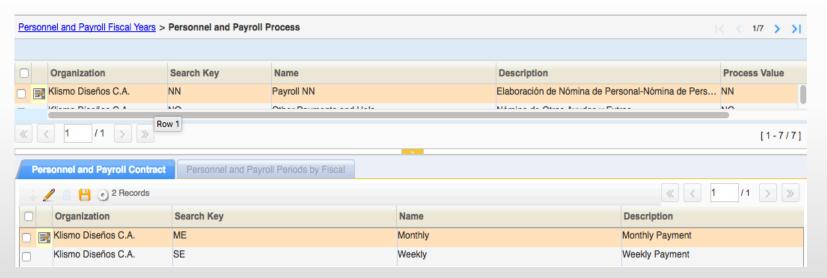


iDempiere World Conference 2019



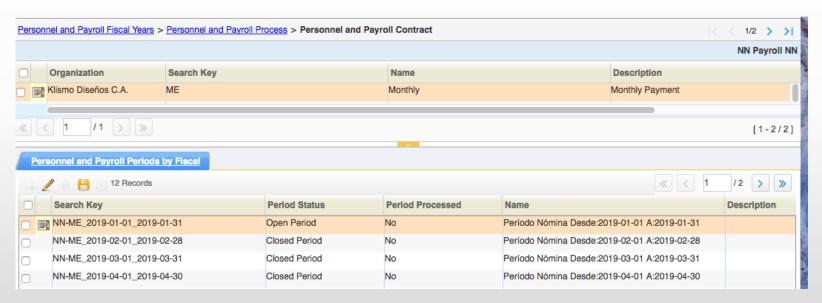


## Payroll Periods (2)



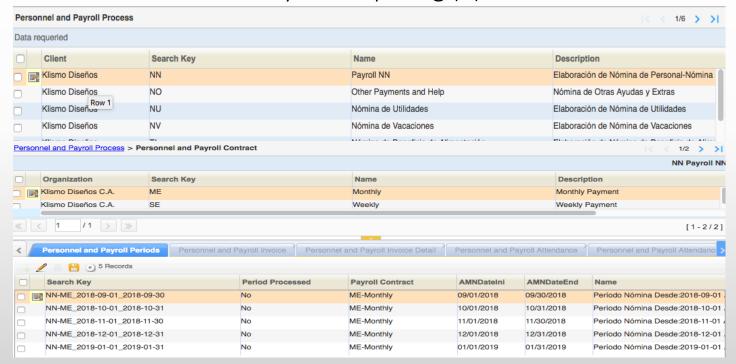


### Payroll Periods (3)





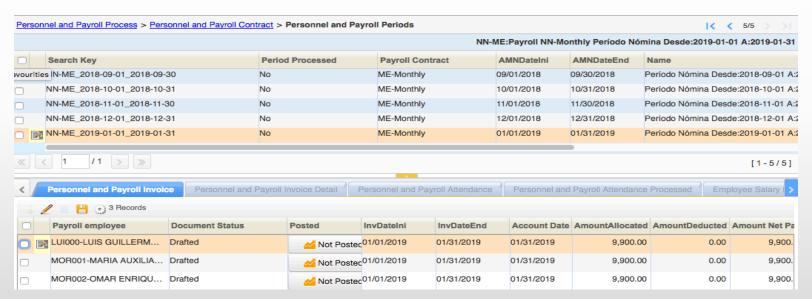
#### Payroll Preparing (1)





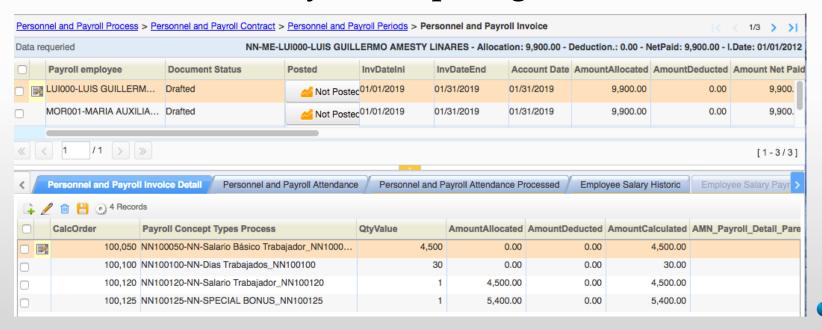


### Payroll Preparing (2)



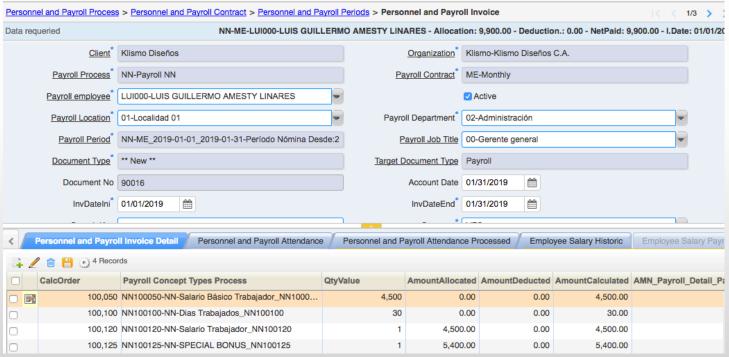


### Payroll Preparing (3)





### Payroll Preparing (4)



6



# Reports

### Reports # 1

Client: Klismo

- Payroll Reports
- Payroll Receipts
- Attendance
- Loans
- Social Benefits statements
- Accounting
- Withholdings and Taxes
- Food Ticket
- Employees
- Other Benefits
- Other Taxes and Withholdings





# Reports

#### Reports # 1 (Application Dictionary)

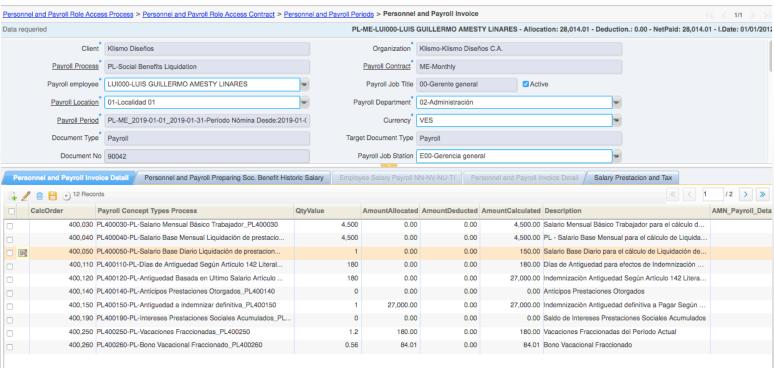
Process: Amper Payroll List Report Landscape Parameters: (Based on Role-Access Model)

rarameters. (Basea of Notes Notes)							
DB Column Name	AMN_Contract_ID	System Element	Q				
Reference	Table	Reference Key	AMN_Contract_ID				
Value Forma	t	<u>Dynamic Validation</u>	AMN_Contract_ID - By Role Access if Process=@AMN_Prot				
AMN_Contract.AMN_Contract_ID in (SELECT DISTINCT AMN_Contract_ID FROM AMN_Role_Access WHERE AMN_Role_Access.ad_role_id = @AD_Role_ID@ AND AMN_Role_Access.AMN_Process_ID = @AMN_Process_ID@ )							
DB Column Name* A	MN_Process_ID	System Element		Q			
Reference*	able	Reference Key	AMN_Process_ID				
Value Format		Dynamic Validation	AMN_Process_ID - By Role Access	-			
AMN_Process.AMN_Process_ID in (SELECT DISTINCT AMN_Process_ID FROM AMN_Role_Access WHERE AMN_Role_Access.ad_role_id = @AD_Role_ID@)							
Reference*	Table	Reference Key	AMN_Period_ID order DESC	•			
Value Format		Dynamic Validation	AMN_Period_ID Open Only	•			

AMN\_Period.AMN\_Contract\_ID = @AMN\_Contract\_ID@ AND AMN\_Period.AMN\_Process\_ID = @AMN\_Process\_ID@ AND AMN\_Period.AMN\_Period\_Status = '0'



### Payroll Preparing Social Benefits(1)



### Payroll Preparing Social Benefits (2)

CalcOrder	Search Key	Default Value	Formula
400.030	PLSALMEN	SBMIN	CN
400.040	SALMENLIQ	if (SBMIN > DV_SALARYLAST ) {(SBMIN ; } else { DV_SALARYLAST ; }	CN
400.050	SALBASLIQ	1	CN * (QT SALMENLIQ / 30)
400.110	ANTIGUEDADC	if ((RV_FDAYS_SERVICE) > 180) { 30 * RV_FYEAR_SERVICE;} else { 5 * ( RV_FDAYS_SERVICE / 30);}	CN
400.120	ANTIGUEDADCM	if ((RV_FDAYS_SERVICE) > 180) { 30 * RV_FYEAR_SERVICE;} else { 5 * ( RV_FDAYS_SERVICE / 30);}	RS_SALBASLIQ*CN
400.140	ANTICIPPREST	0	CN
400.150	ANTIGUEDADPA	1	RS_ANTIGUEDADCM * CN
400.190	INTERESESPS	0	CN
400.250	VACACFRAC	RV_FVACAC190	RS_SALBASLIQ*CN
400.260	BONOVACACF	RV_FVACAC190BON	RS_SALBASLIQ*CN
400.300	UTILFRAC	60 * ( RV_FDAYS_THISYEAR / 360 )	CN * (QT_PLSALMEN/30)
400.800	ANTICIPUTILI	0	CN





### Payroll Preparing Social Benefits (3)

Social benefit Final Payment: Based on Employee Income Date Using RULES

RV FVACAC190 Vacation Fraction

RV FVACAC190BON Vacation Bonus Fraction

RV FDAYSFRAC Days elapsed from last employee anniversary RV FDAYS THISYEAR Days elapsed

from 01/01/XXXX

RV\_FYEAR\_SERVICE Number of years of service from Employee RV FDAYS SERVICE Number of days of service from Employee





#### Payroll Preparing Social Benefits (5)

RV FVACAC190

Vacation Fraction

RV FVACAC190BON

Vacation Bonus Fraction

```
// Rule: RV_FVACAC190
       import org.compiere.model.*
       import org.compiere.util.*;
       import java.math.*;
       import java.sql.Timestamp;
       import java.text.DateFormat
       import java.text.SimpleDateFormat:
       import java.util.ArrayList;
       import java.util.Calendar;
       import java.util.GregorianCalendar;
       import java.util.Properties;
       import java.math.BigDecimal;
       // BEGIN
       Timestamp start =A_EmployeeIncomeDate;
       Timestamp end =A_InvDateEnd;
       // Callni Employee Income date
       Calendar callni = Calendar.getInstance();
       callni.setTime(start);
       int firstYearValue = callni.get(Calendar.YEAR);
       int firstDayValue = callni.get(Calendar.DAY_OF_YEAR);
Script // CalEnd A_InvDateEnd
      Calendar calEnd = Calendar.getInstance();
       calEnd.setTime(end);
       int secondYearValue = calEnd.get(Calendar.YEAR);
       int secondDayValue = calEnd.get(Calendar.DAY_OF_YEAR);
       // Calc Diff in days elapsed
       int diffdayslast = firstDayValue - secondDayValue;
      if (diffdayslast < 0 )
           diffdayslast = secondDayValue - firstDayValue;
       // Calc Fraction in Big Decimal
      int i360 = 360:
      int i15 = 15;
       int iZero = 0;
       BigDecimal DBVacacDays = new BigDecimal(iZero);
      BigDecimal DBdiffDays= new BigDecimal(diffdayslast);
BigDecimal BD360 = new BigDecimal(i360);
       BigDecimal BD15 = new BigDecimal(i15);
       BigDecimal DBVacacDays = DBdiffDays.divide(BD360,2, BigDecimal.ROUND_HALF_UP);
       DBVacacDays = DBVacacDays.multiply(BD15);
       DBVacacDays.setScale(2, RoundingMode.HALF_UP);
       return DBVacacDays;
```

```
import java.text.SimpleDateFormat;
import java.util.ArrayList;
                                                                        import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.Properties;
                                                                               import java.math.BigDecimal;
// BEGIN
                                                                        // BEGIN
Timestamp start =A_EmployeeIncomeDate;
Timestamp end =A_InvDateEnd;
// Callni Employee Income date
Calendar callni = Calendar.getInstance();
                                                               calinia daini - Zaenda gusinsance(), caliniast Time(start); int first YearValue = calini.get(Calendar.PAR); int first YearValue = calini.get(Calendar.DAY_OF_YEAR); // Calendar_caliend = Calendar_caliend 
int second/Year/Value = calEnd.get(Calendar.YEAR);
int second/Day/Value = calEnd.get(Calendar.DAY_OF_YEAR);
int second/Day/Value = calEnd.get(Calendar.DAY_OF_YEAR);
if Calendrary = calEnd.get(Calendar.DAY_OF_YEAR);
int diffdayslast = firstDay/Value - second/Day/Value;
if (diffdayslast < 0)
                                                                     diffdayslast = secondDayValue - firstDayValue;
// Calculates No of years
int diffyears = secondVearValue - firstYearValue;
// Year Validation > 180 means one year
                                                                  if (diffyears < 0 )
DByearService = BigDecimal.ZERO;
                                                                  Deyearservice = BigDecimal.ZERC;
if (diffyears > 0 ) {
            // IF Fragment of Deyearser than 180 then add a year
            if (diffyears > 180)
            diffyears = diffyears + 1;
            DByearService = new BigDecimal(diffyears);

                                                                  // Calculates No of years (Max to 15)
if (diffyears > 15 )
diffyears=15;
                                                                           // Final Big Decimal Calc
                                                                     // Final Big Decimal Calc
BigDecimal BOdiffdaysia-new BigDecimal(diffyears);
BigDecimal BOdiffdayslast= new BigDecimal(diffdayslast);
int 380 = 380;
BigDecimal BD380 = new BigDecimal(380);
BigDecimal BD380 = new BigDecimal(380);
BigDecimal BD380 = new BigDecimal(380);
BigDecimal BD890n/vaceFrac = DB0fidfdayslast.divide(BD380.2, BigDecimalROUND_HALF_UP);
DBB0n/vaceFrac = DBB0n/vaceFrac = DB0n/vaceFrac = DB0n/vac
                                                                        DBBonVacacFrac.setScale(2, RoundingMode.HALF_UP);
return DBBonVacacFrac;
```





#### Payroll Preparing Social Benefits (6)

#### RV\_FDAYSFRAC

```
Days elapsed from last employee anniversary
```

```
// RV FDAYSFRAC
       import org.compiere.model.*;
       import org.compiere.util.*;
       import java.math.*;
       import java.sql.Timestamp
       import java.text.DateFormat;
       import java.text.SimpleDateFormat;
       import java.util.ArrayList;
       import java.util.Calendar,
       import java.util.GregorianCalendar;
       import java.util.Properties;
       import java.math.BigDecimal;
       // BEGIN
       Timestamp start =A_EmployeeIncomeDate;
       Timestamp end = A InvDateEnd;
       // Callni Employee Income date
       Calendar callni = Calendar.getInstance();
       callni.setTime(start);
Script int firstYearValue = callni.get(Calendar.YEAR);
       int firstDayValue = callni.get(Calendar.DAY_OF_YEAR);
       // CalEnd A_InvDateEnd
       Calendar calEnd = Calendar.getInstance();
       calEnd.setTime(end);
       int secondYearValue = calEnd.get(Calendar.YEAR);
       int secondDayValue = calEnd.get(Calendar.DAY OF YEAR);
       // Calc Diff in days elapsed
       int diffdayslast = firstDayValue - secondDayValue;
       if (diffdayslast < 0)
            diffdayslast = secondDayValue - firstDayValue;
       BigDecimal DBdiffDays = new BigDecimal(diffdayslast);
       DBdiffDays.setScale(2, RoundingMode.HALF_UP);
       return DBdiffDays;
```

# RV\_FDAYS\_THISYEAR Days elapsed from 01/01/XXXX

```
// RV FDAYS THISYEAR
       import org.compiere.model.*;
       import org.compiere.util.*;
       import java.math.*;
       import java.sql.Timestamp;
                                                       Dynamic Java
       import java.text.DateFormat;
       import java.text.SimpleDateFormat;
       import java.util.ArrayList;
       import java.util.Calendar;
       import java.util.GregorianCalendar;
       import java.util.Properties:
       import java.math.BigDecimal;
       // BEGIN
       Timestamp end =A InvDateEnd:
       // Callni Fist day of YEAR
       Calendar callni = Calendar.getInstance();
       callni.set(Calendar.DAY_OF_YEAR, 1);
       // CalEnd A_InvDateEnd
Script Calendar calEnd = Calendar.getInstance();
       calEnd.setTime(end);
       int secondYearValue = calEnd.get(Calendar.YEAR);
       int secondDayValue = calEnd.get(Calendar.DAY_OF_YEAR);
       // Calculates Days from Begining of the year
       int firstDayofYear = callni.get(Calendar.DAY OF YEAR);
       int diffdayslast = secondDayValue - firstDayofYear;
       if (diffdayslast < 0)
         diffdayslast=0;
       BigDecimal DBdiffDays = new BigDecimal(diffdayslast);
       DBdiffDays.setScale(2, RoundingMode.HALF_UP);
       return DBdiffDays;
```





# Plugin Code

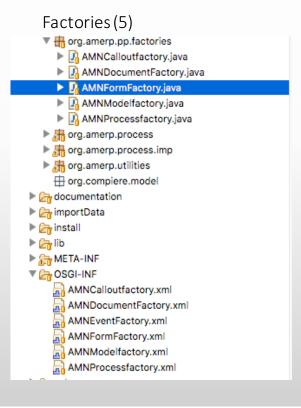
#### Code Structure (1)

#### Views (35 Tables) ▼ 3 org.amerp.amnmodel ▶ IRI\_AMN\_Concept\_Types\_Acct.java ▶ I I\_AMN\_Concept\_Types\_Contract.java ► RI\_AMN\_Concept\_Types\_Proc.java ► R I\_AMN\_Concept\_Types.java ► RI\_AMN\_Concept\_Uom.java ► R I\_AMN\_Concept.java ► RI\_AMN\_Contract.java ▶ R I\_AMN\_Department.java ► R I\_AMN\_Dependent\_type.java ► RI\_AMN\_Dependent.java ► IRI AMN Employee Salary.iava ► RI\_AMN\_Employee\_Tax.java ► RI\_AMN\_Employee.java ▶ R LAMN LEmployee Salary.java ▶ RI\_AMN\_Jobstation.java ▶ R I\_AMN\_Jobtitle.java ► RI\_AMN\_Jobunit.java ► RI\_AMN\_Location.java I\_AMN\_Payroll\_Assist\_Proc.java ► RI\_AMN\_Payroll\_Assist.java ► RI\_AMN\_Payroll\_Deferred.java ► R I\_AMN\_Payroll\_Detail.java ► RI\_AMN\_Payroll\_Historic.java ► RI\_AMN\_Payroll\_Lot.java

► ☐ I\_AMN\_Payroll.java

► ☐ I\_AMN\_Period\_Assist.java

MI AMM Deriod inva



#### Callouts ▼ A org.amerp.callouts AMN\_Concept\_callout.java ▶ MAN\_Concept\_Types\_callout.java AMN\_Concept\_Types\_Contract\_callout.java ▶ ☐ AMN\_Concept\_Types\_Proc\_callout.java ► AMN\_Employee\_Tax\_callout.java AMN\_Payroll\_Assist\_BioCode\_callout.java AMN\_Payroll\_Assist\_callout.java AMN\_Payroll\_Assist\_Proc\_callout.java ▶ AMN\_Payroll\_callout.java AMN\_Payroll\_Dates\_callout.java ▶ AMN\_Payroll\_Deferred\_callout.java AMN\_Payroll\_Detail\_callout.java AMN\_Payroll\_Detail\_Qty\_callout.java ▶ MAMN\_Payroll\_Lot\_callout.java AMN\_Period\_Assist\_callout.java ▶ AMN\_Period\_callout.java ▶ AMN\_Process\_callout.java AMN\_Role\_Access\_callout.java ScriptEngineTest.java



# Plugin Code

#### SQL Functions (6)

#### 

#### **Application Dictionary:**

Windows (33)

Tables and Views (40)

Processes References Validation Rules

Messages

#### SQL Views (5)



#### **JASPERSOFT Reports:**

Employees (2)

Payroll (10)

Receipt reports (12)

Attendance (3)

Loans (2)

Social Benefit (3)

Accounting Control (4) Withholding Taxes (1)

Food Ticket (2)

Others (5)





# Plugin Code

#### AmerpPayrollCall.java Class

- S PayrollEvaluation(Properties, int, int, boolean): BigDecimal
- S PayrollEvaluationArrayCalculate(Properties, int): void
- S FormulaEvaluationScript(int, String, String, BigDecimal, BigDecimal, B
- S logVariablesShow(): String
- S logRulesShow(): String
- ▼ 🛁 org.amerpsoft.personnelpayroll [release-5.1 ↑51 :]
  - ▼ # src
    - ▼ Æ g org.amerp.amnmodel
      - ▼ MAMN\_Payroll\_Detail
        - createAmnPayrollDetail(Properties, Locale, int, int, int, int, int, int, String)
    - ▼ ♣ org.amerp.callouts
      - ▼ 🕝 a AMN\_Payroll\_Detail\_callout
        - a start(Properties, int, GridTab, GridField, Object, Object) (3 matches)
      - ▼ 🕞 AMN\_Payroll\_Detail\_Qty\_callout
        - start(Properties, int, GridTab, GridField, Object, Object)
    - ▼ ♣ org.amerp.utilities
    - ▼ 🚱 🛮 AmerpPayrollCalc
      - S PayrollEvaluationArrayCalculate(Properties, int) (2 matches)

**Process** 

Callout

Internal



# Thank you



Merci à vous Il a été un plaisir Des questions?

Auyantepuy, Venezuela

