



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Automatización del proceso de ventas de una empresa
automotriz en SAP usando metodología Scrum**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Cristian Daniel SARAZU COTRINA

ASESOR

Cesar Alberto MOLINA NEYRA

Lima, Perú

2022



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Sarazu, C. (2022). *Automatización del proceso de ventas de una empresa automotriz en SAP usando metodología Scrum*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Cristian Daniel Sarazu Cotrina
Tipo de documento de identidad	DNI
Número de documento de identidad	45810639
URL de ORCID	https://orcid.org/0000-0001-7603-706X
Datos de asesor	
Nombres y apellidos	Cesar Alberto Molina Neyra
Tipo de documento de identidad	DNI
Número de documento de identidad	40553679
URL de ORCID	https://orcid.org/0000-0002-8018-7973
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Santiago Domingo Moquillaza Henríquez
Tipo de documento	DNI
Número de documento de identidad	08280889
Miembro del jurado 1	
Nombres y apellidos	Carlos Edmundo Navarro Depaz
Tipo de documento	DNI
Número de documento de identidad	08482690
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Perú, Lima, Lurín, Ant. Panamericana Sur Km 38.2, Lurín Latitud: -12.29027 Longitud: -76.85590
Año o rango de años en que se realizó la investigación	2021 – 2022
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 19:00 horas del día 15 de enero del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Moquillaza Henríquez Santiago Domingo (Presidente), Dr. Navarro Depaz Carlos Edmundo (Miembro) y el Ing. Molina Neyra Cesar Alberto (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/jjy-yahj-fza>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“AUTOMATIZACIÓN DEL PROCESO DE VENTAS DE UNA EMPRESA AUTOMOTRIZ EN SAP USANDO METODOLOGÍA SCRUM”**, por el Bachiller **Sarazu Cotrina Cristian Daniel**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **16 DIECISEIS**.

A continuación el Presidente de Jurados el Mg. Moquillaza Henríquez Santiago Domingo, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 19:50 horas, se levantó la sesión.

Presidente

Mg. Moquillaza Henríquez Santiago Domingo

Miembro

Dr. Navarro Depaz Carlos Edmundo

Miembro Asesor

Ing. Molina Neyra Cesar Alberto

DEDICATORIA

El presente trabajo es dedicado a mi hijo, que con su nacimiento cambio mi vida y me inspiro ser una mejor persona y un buen profesional para ser su ejemplo a seguir.

AGRADECIMIENTO

A mis padres:

Por los valores que me inculcaron y por su confianza en el camino para cumplir con mis objetivos

A la Universidad:

Por las experiencias vividas y enseñanzas que me dejó a lo largo de mi permanencia

A mi asesor:

Por los consejos, observaciones y recomendaciones
Para la elaboración del presente trabajo

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
AUTOMATIZACIÓN DEL PROCESO DE VENTAS DE UNA EMPRESA
AUTOMOTRIZ EN SAP USANDO METODOLOGÍA SCRUM

Autor: Sarazu Cotrina, Cristian Daniel
Asesor: Molina Neyra, Cesar Alberto
Titulo: Trabajo de Suficiencia Profesional para optar el Título Profesional
de Ingeniera de Sistemas
Fecha: Enero 2022

RESUMEN

El presente trabajo de suficiencia profesional explica la automatización del proceso de ventas de una empresa automotriz que usa SAP ERP como software para todos sus procesos, cuyo objetivo principal es aumentar las ventas y mejorar su proceso de ventas.

La metodología usada para este trabajo fue SCRUM debido a que el alcance del proyecto no estaba completamente definido y SCRUM permite la entrega de producto funcional mientras se mejora o se planifica mejoras en el mismo por cada entrega. La automatización del proceso de ventas en SAP ERP trajo consigo un aumento del 232% en las ventas, una disminución en el tiempo que tomaba la creación de las cotizaciones para los clientes antes duraba 3 días la generación de estas y luego de la implementación del proyecto solo demoraba 10 minutos, generación automática de pedidos, ordenes de fabricación y facturas, así como una disminución en la pérdida de los clientes potenciales.

Palabras clave: *SCRUM, SAP ERP, Proceso de ventas, Cotizaciones de ventas.*

NATIONAL MAJOR UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS ENGINEERING AND INFORMATICA
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

Automation of the sales process of an automotive company in SAP using Scrum

methodology

Author: Sarazu Cotrina, Cristian Daniel

Advisor: Molina Neyra, Cesar Alberto

Title: Professional Sufficiency Work for opt for the Professional title of
Systems Engineer

Date: Enero 2022

ABSTRACT

The present professional sufficiency work explains the automation of the sales process of an automotive company that uses SAP ERP as software for all its processes, the main objective of the company is to increase sales and improve its sales process.

The methodology used for this work was SCRUM because the scope of the project was not fully defined and SCRUM allows the delivery of functional product while improving or planning improvements in it for each delivery. The automation of the sales process in SAP ERP brought with it a 232% increase in sales, a decrease in the time it took to create quotes for customers before the generation of these lasted 3 days and after the implementation of the project only took 10 minutes, automatic generation of orders, manufacturing orders and invoices, as well as a decrease in the loss of potential customers.

Keywords: SCRUM, SAP ERP, Sales Process, Sales Quotes.

Tabla de Contenido

CAPITULO I. TRAYECTORIA PROFESIONAL	1
CAPITULO II. CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	4
2.1 Empresa	4
2.2. Visión	4
“Ser la empresa líder en consultoría tecnológica en el Perú aumentan la confianza y nivel de satisfacción de nuestros clientes” (Soluciones Empresariales SAP EIRL, s.f.).	4
2.3. Misión	4
2.4. Organización de la Empresa	5
2.5. Área, Cargo y funciones desempeñadas	5
2.6 Experiencia Profesional Realizada en la Organización.	5
3.1. Situación Problemática.	6
3.1.1 Definición del Problema.....	6
3.1.1.1. Problema Principal.	12
3.2. Solución	13
3.2.2 Alcance	14
3.2.3 Etapas y Metodología Usada	14
3.2.4 Fundamentos Utilizados	15
3.2.5. Implementación de las Áreas de procesos y buenas practicas.....	18
3.3. EVALUACIÓN	23
3.3.1. Evaluación económica.....	23
3.3.2. Beneficios Obtenidos	23
CAPITULO IV. REFLEXIONE CRITICA DE LA EXPERIENCIA.....	25
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES	26
5.2. Recomendaciones.	26
5.3. Fuentes de Información.	27
5.4. Glosario	28
ANEXOS	29
ANEXO-01	30

INDICE DE FIGURAS

Figura 1. Organigrama de la empresa.....	5
Figura 2. Modelo de cotización de Buses usado anteriormente	7
Figura 3. Cotización y características del equipo comprado en SAP ERP, la cotización tiene que ser llenada en base al Excel que el vendedor a cotizado con el cliente.	8
Figura 4. Transacción de pedido de ventas en SAP ERP	11
Figura 5. Creación de la orden de Fabricación en SAP ERP	11
Figura 6. Entrega de salida SAP ERP.....	12
Figura 7. Factura de ventas en SAP ERP	12
Figura 8. Proceso Scrum (SCRUM ORG, 2021)	15
Figura 9. Valores de Scrum (Schwabe & Sutherland, 2020).....	16
Figura 10. Diseño de la base de datos	21
Figura 11 Modelo de cotización	22
Figura 12. Transacción para la creación de factura con referencia a la cotización o pedido	22

INDICE DE TABLAS

Tabla 1 Experiencia Profesional.....	1
Tabla 2 Formación Profesional	3
Tabla 3 Certificaciones	3
Tabla 4 Historias de usuario	18
Tabla 5 Sprint, historias de usuario y tareas realizadas	20
Tabla 6. Costo del proyecto	23
Tabla 7.Tabla de beneficio en la duración de la creación de las cotizaciones	24
Tabla 8. Tabla de beneficio económico obtenido del proyecto (Fuente Empresa automotriz)	24

INTRODUCCION

El presente informe profesional describe la automatización del proceso de ventas de una empresa automotriz que usa SAP ERP como software para sus procesos, esto con el objetivo de aumentar las ventas, reducir el tiempo de duración del proceso de ventas, disminuir el número de potenciales clientes que declinan de una compra debido al tiempo en que demora la creación de una cotización.

El proyecto se originó debido a que se notó que había una gran cantidad de clientes que declinaban de comprar debido a que el modo en que se iniciaba la venta por medio de una cotización tardaba 3 días la aprobación y creación del pedido de ventas 1 día lo cual causaba que el cliente se disgustaba o prefería buscar otras opciones.

El informe se desarrolló con la siguiente estructura:

En el CAPITULO I se detalla cronológicamente la trayectoria profesional del autor, las funciones, principales logros, aprendizajes y experiencia adquirida que se tuvo en cada posición laboral, así como también los conocimientos adquiridos a través de las diversas capacitaciones, cursos y diplomados realizados por el autor.

En el CAPITULO II se resume parte de las responsabilidades desempeñadas en la organización, la visión, misión y la estructura de la organización.

En el CAPITULO III se describe el principal problema que atravesó la organización y se indica la solución para la misma, en donde se describe las etapas, metodologías utilizadas y la evaluación económica.

En el CAPITULO IV se indica la reflexión crítica de la experiencia realizada luego de implementar la solución en la organización.

En el CAPITULO V se mencionan las conclusiones del informe y recomendaciones para posteriores trabajos.

CAPITULO I. TRAYECTORIA PROFESIONAL

El autor actualmente tiene el grado de bachiller en Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos. Presenta Habilidades en liderazgo y gestión de Proyectos SAP usando la Metodología SCRUM.

También cuenta con habilidades interpersonales y blandas para trabajar en equipo, resolver problemas y demostrar influencia e impacto.

Adicionalmente Poseo Certificaciones en Los Módulos SAP FI-MM-ABAP, Certificación en SCRUM PSM-I.

A continuación, detallo experiencia profesional.

Tabla 1 Experiencia Profesional

Entidad	Experiencia			
	Área	cargo	fecha inicio	Fecha de fin
ABSIDE	Sistemas Consultoría	Jefe de Proyectos	Oct-21	Actualidad
	Jefe de Proyectos en la Implementación SAP S4/HANA para La Fabril Funciones: * Formar el Equipo de Trabajo * Gestionar los Costos, Alcance y tiempos de los proyectos * Hacer informes de desempeño de trabajo * Implementar los estándares para los desarrollos * Elaborar las Plantillas para la documentación Técnica y Funcional			
Entidad	Experiencia			
	Área	cargo	fecha inicio	Fecha de fin
TATA Consulting	Sistemas Consultoría	Consultor SAP	Ago-21	Actualidad
	Consultor en la Implementación SAP S4/HANA en la empresa QROMA Funciones: * Formar el Equipo de Trabajo * Gestionar los Costos, Alcance y tiempos de los proyectos * Hacer informes de desempeño de trabajo * Implementar los estándares para los desarrollos * Elaborar las Plantillas para la documentación Técnica y Funcional			

Entidad	Experiencia			
	Area	cargo	fecha inicio	Fecha de fin
Soluciones Empresariales SAP	Sistemas Consultoría	Jefe de Proyectos	Mar-20	Ago-21
	<p>Jefe de Proyectos en la Implementación de y mejora a los módulos del Sistema SAP R3/ SAP FIORI de los Clientes (Modosa, Repsol, San Martin)</p> <p>Funciones:</p> <ul style="list-style-type: none"> * Analizar la Viabilidad del Proyecto * Formar el Equipo de Trabajo * Gestionar los Costos, Alcance y tiempos de los proyectos * Hacer informes de desempeño de trabajo * Implementar los estándares para los desarrollos * Elaborar las Plantillas para la documentación Técnica y Funcional 			
Experis	Sistemas Consultoría	Sub Líder SAP	Ago-18	Mar-20
	<p>Sub Líder Técnico en la Implementación de SAP S4/HANA en la Empresa Ferreyros</p> <p>Funciones:</p> <ul style="list-style-type: none"> * Asignar requerimientos al personal a cargo * Capacitar a los consultores nuevos * Desarrollar programas a Medida según requerimientos Funcionales * Realizar las pruebas Unitarias de los desarrollos * Estimar tiempo de desarrollo de los requerimientos 			
Soluciones Empresariales SAP	Sistemas Consultoría	Consultor ABAP	Ene-15	Jul-18
	<p>Consultor ABAP en proyectos de mejoras continuas para el Sistema SAP R3 en los clientes (MODASA)</p> <p>Funciones:</p> <ul style="list-style-type: none"> * Desarrollar programas a medida, según solicitud del usuario * Hacer las Pruebas funcionales 			

Tabla 2 Formación Profesional

Formación Académica	Institución	Periodo
Grado de Bachiller en Ingeniería de sistemas	Universidad Nacional Mayor de San Marcos	2008-2015

Tabla 3 Certificaciones

Certificación	Institución que Emite la certificación	Año
The Professional Scrum Master I (PSM I)	SCRUM.ORG	2020
SAP Certified Application Associate - SAP S/4HANA Sourcing and Procurement - Upskilling for ERP Experts	SAP	2020
SAP Certified Application Associate - SAP S/4HANA for Financial Accounting Associates (SAP S/4HANA 1809)	SAP	2020
SAP Certified Development Associate - ABAP with SAP NetWeaver 7.50	SAP	2020

CAPITULO II. CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1 Empresa

Soluciones Empresariales SAP es una empresa que inicio sus labores en Perú el año 2016 dedicándose al rubro de tecnologías de información y consultoría.

Información de la empresa

Razón Social: Soluciones Empresariales SAP EIRL.

Ruc: 20600969529

Dirección legal: Av. Malecón Mz. e Lote. 37 Sauces.

Distrito / ciudad: Puente Piedra.

Soluciones empresariales SAP es especialista en brindar soluciones de Implementaciones, Mejoras continuas y consultoría sobre el Sistema SAP.

2.2. Visión

“Ser la empresa líder en consultoría tecnológica en el Perú aumentan la confianza y nivel de satisfacción de nuestros clientes” (Soluciones Empresariales SAP EIRL, s.f.).

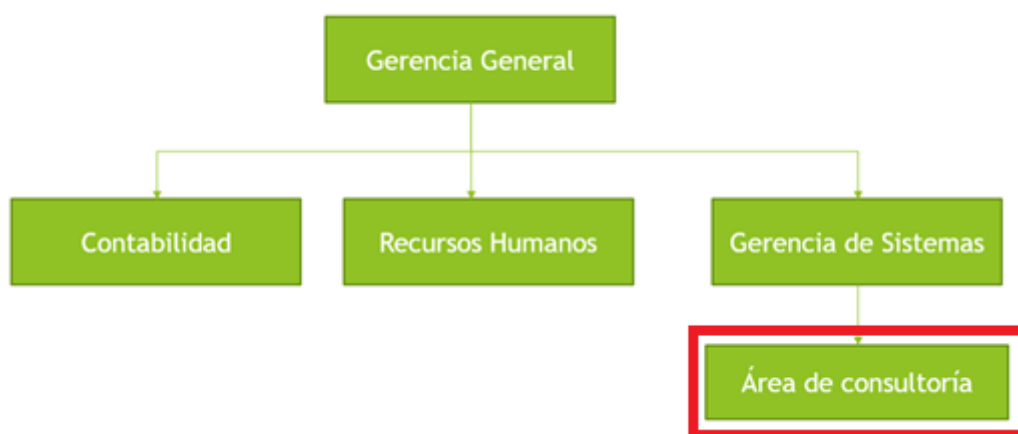
2.3. Misión

“Brindar soluciones tecnológicas, estratégicas y organizativas dar a nuestros clientes ventajas competitivas con respecto a su competencia” (Soluciones Empresariales SAP EIRL, s.f.).

2.4. Organización de la Empresa

A continuación, se presenta el Organigrama de la Empresa.

Figura 1. Organigrama de la empresa



2.5. Área, Cargo y funciones desempeñadas

El autor de este informe a lo largo de su experiencia profesional, ocupó el rol de jefe de Proyectos SAP, participando en diversos proyectos de implementación, mejoras continuas y Roll Out para diversos clientes importantes, trabajando en el área de consultoría, tuvo un crecimiento profesional iniciando como consultor técnico, ascendiendo a líder técnico y terminando como jefe de proyectos SAP.

2.6 Experiencia Profesional Realizada en la Organización.

El autor del Informe como jefe de Proyectos SAP participó en los siguientes proyectos:

- Proyecto de Implementación de módulo de cotización y venta de buses para empresa automotriz.
- Proyectos de Mejora al Portal de Proveedores de Empresa Constructora.
- Proyecto de Mejora a la Facturación Electrónica de Empresa Industrial
- Proyecto de Mejora continua en empresa Automotriz
- Proyecto Vendedores de campo para empresa Automotriz.

CAPITULO III – ACTIVIDADES DESARROLLADAS

3.1. Situación Problemática.

3.1.1 Definición del Problema

El ERP SAP si bien es un Software que cumple con la mayoría de las necesidades de las empresas al tener integrado todas las áreas de las mismas, ya sea Contabilidad, ventas, logística, producción, etc. Las empresas siempre buscan personalizaciones adicionales, ajustes o implementación de módulos que ayuden a cumplir con lo específico de su rubro.

Todas las empresas que tienen Implementado el ERP SAP trabajan con consultoras las cuales le brindan el soporte para sus mejoras e implementaciones de proyectos que involucren al ERP SAP.

La empresa de rubro Automotriz dedicada a la fabricación y venta de buses y grupos electrógenos, tiene como parte de su organización vendedores de campo, los cuales tenían el objetivo de buscar clientes potenciales, brindarles información y cotización de los buses, pero en ese momento no se contaba con un sistema el cual pueda facilitar la creación de una cotización la cual se adapte a las necesidades del negocio, simplemente el vendedor manejaba un Excel con los datos de buses y era poco personalizable respecto de lo que quería el cliente lo cual ocasionaba perdida de los potenciales clientes y retrasos en los cierres de los tratos con los clientes, además del tema burocrático aprobaciones de gerencia impedía un ágil flujo del proceso de cotización y venta de los buses lo cual cansaba a los clientes y estos desistían de sus compras.

El modelo de la cotización que se visualiza en la Figura 2, se usaba para recoger los requerimientos de los clientes y elaborar una cotización con el precio final de venta, esto luego de unos días de análisis con los encargados de producción y ventas para calcular costos y márgenes, posteriormente la cotización debería ser aprobada por el gerente de ventas, esto causaba que el proceso de ventas sea muy lento ya que involucraba análisis y autorizaciones aparte de manejar toda la documentación física de un área a otra.

Figura 2. Modelo de cotización de Buses usado anteriormente

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
TIPO	AÑO	MODELO	TIPO	clave	MODELO	COMBUSTI	FORMULA			COSTO_MANO_OBRA	TITULO	SUBTITULO	características y valores		VALOR	PRECIO_TITULO	MONED
BUS	XXX	XXXXX	XXX	HINO	FC BUS	DIESEL	4X2			28,892.96	DIMENSIONES				9100 mm 2400 mm 3100 mm		0.00 PEN
											CHASIS						PEN
												MARCA-MODELO-CHASIS COMBUSTIBLE FORMULA RODANTE			HINO FC BUS Diesel 4X2		
											CARROCERIA						PEN
												ASIENTOS					
													CANTIDAD TIPO FORRADO FORRO ESPALDAR DECORATIVO EN VINIL PORTA-REVISTERO CINTURÓN DE SEGURIDAD BELCRO FUNDAS DE CABEZAL RECLINACIONES		26 Soft Junior Tela Navallada Tapizón No Lija 3 puntos retráctil No Si 3		
												ASIENTO PILOTO					
													TIPO FORRADO		Grammer Original		
												ASIENTOS COPILOTO					
													TIPO FORRADO CINTURON DE SEGURIDAD		Plegable Igual a asientos de salón 3 puntos retráctil		
												SISTEMAS INTERNOS					PEN
													BAÑO EQUIPAMIENTO DE SERVICIO PAQUETERA		No No		
													FORRO LATERAL FORRO CENTRAL LIGAPORTAEQUIPAJE		Vinil Texturizado Vinil Pranna Si		
												FORRADO DE PISO			Pisobus		
												CORTINAS		APLICACIÓN			
													CORTINAS LATERALES				
													TIPO SUJECION		Tela Ojal Superior Si Mampara		
												TAPASOL DE CONDUCTOR MAMPARA					
												SISTEMAS EXTERNOS					PEN
													MODELOS DE ESPEJO POSICION DE RUEDA DE REPUESTO		Apolo 9 Central		
												CLIMATIZACIÓN					PEN
													RENOVADORES DE AIRE AIRE ACÓNCONDICIONADO PASAJERO CALEFACCION CLARABOYAS DESEMPAÑADOR AIRE FORZADO		2 No Si 1 Aire caliente No		

Figura 3. Cotización y características del equipo comprado en SAP ERP, la cotización tiene que ser llenada en base al Excel que el vendedor a cotizado con el cliente.

Visualizar Cotización Equipo 2300000132: Resumen

Pedidos

Cotización Equipo	2300000132	Valor neto	63,025.21	USD
Solicitante	99999	CUSCOCOACHING EIRL / NRO. R-19 APV LOS NOGALES / CUSCO		
Destinat.mó.	99999	CUSCOCOACHING EIRL / NRO. R-19 APV LOS NOGALES / CUSCO		
Nº ped.cliente	BUS VOLKSWAGEN	Fecha de pedido	16.06.2009	

Ventas | **Resumen de posiciones** | Detalle posición | Solicitante | Aprovisionamiento | Expedición | Motivo de rechazo

Válido de	16.06.2009	Valdez a	
Fe.pref.entrg.	D 16.06.2009	Valor esper.ped	44,117.65 USD

Pos.	Material	Cantidad de pedido	UM	AltPos	Denominación
10	E30T00003	1.000	UN		0 BUS APOLO TURISMO VI VW 9.150 OD

Modificar Cotización Equipo 2300000132: Valoración de característica

Solicitante	99999	CUSCOCOACHING EIRL
Material	E30T00003	BUS APOLO TURISMO VI VW 9.150 OD
Cantidad	1.000	UN
Fecha pref.entrega	16.06.2009	Posición 10

Valoración de la característica

Denom.característica	Valor caract.	In...
TIPO DE COMBUSTIBLE VA	DIESEL	
TIPO DE TANQUE VA	POLICARBONATO	
CAPACIDAD TANQUE VA	150 L	
BANO VA	POSTERIOR	
UBICACIÓN BANO VA	POSTERIOR	
CANTIDAD ASIENTO VA	28	
CANTIDAD ASIENTOS FACTUR VA	30	
TIPO ASIENTO PILOTO VA	PILOTO GRAMMER	
TIPO ASIENTO GUIA VA	NO	
CANTIDAD PORTAVASOS VA	28	
CANTIDAD DE PORTAREV VA	25	
TIPO DE MAMPARA VA	CUBREPIERNAS	

Una vez que la cotización a sido ingresado al Sistema (SAP ERP) el gerente de ventas se encarga de revisar la cotización y aprobar o rechazar dicha cotización, si se aprueba la cotización se procede a crear un pedido de ventas. en la figura 4 se muestra la ventana de pedido de ventas que brinda SAP ERP de forma estándar para ingresar los datos del pedido.

Figura 4. Transacción de pedido de ventas en SAP ERP

Visualizar Cotización Equipo 2300000132: Resumen

Ventas **Resumen de posiciones** Detalle posición Solicitante Aprovisionamiento Expedición Motivo de rechazo

Cotización Equipo: 2300000132 Valor neto: 63,025.21 USD
 Solicitante: 99999 CUSCOCOACHING FIRL / NRO. R-19 APV LOS NOGALES / CUSCO
 Destinat.mcia.: 99999 CUSCOCOACHING FIRL / NRO. R-19 APV LOS NOGALES / CUSCO
 Nº ped.ciente: BUS VOLKSWAGEN Fecha de pedido: 16.06.2009

Válido de: 16.06.2009 Validez a:
 Fe.pref.entrg.: D 16.06.2009 Valor esper.ped: 44,117.65 USD

Pos.	Material	Cantidad de pedido	UM	AltPos	Denominación
10	30UR0003	1.000	UN		0BUS APOLO TURISMO VI VW 9.150 OD

Posterior a la creación del pedido de ventas, se crea la orden de fabricación del producto para la elaboración y posterior entrega al cliente.

Figura 5. Creación de la orden de Fabricación en SAP ERP

Orden fabricación visualizar: Cabecera

Material Capacidad

Orden: A09012019-01 Cl.: ZBUS
 Material: T30UR0012 B-APOLO URB GAS FAW C/MOTOR FIAT-L1 Ce.: 1102
 Status: ABIE PREC EDET DMNV LIBR NLIQ SUJL

General **Asignación** Entr.mercancías Control Fechas/Ctd. Datos maestros Texto explicativo Gestión

Responsabilidad

Planif.necesidades: P01 MB Lurin-Lima
 Resp.ctrl.producción: BAT Respons Terminados Buses Ate

Centros

Centro de producción: 1102 Centro planificación: 1102
 Área pl.nec.: 1102 MB Lurin-Lima

Asignaciones

Elemento PEP		Lote de inspección	0
Pedido cliente	1300007289	Orden repetitiva	
Nº explos.LMat	10	Orden previsual	1425180
División	1102	Versión fabricación	
Área funcional	CPRO Costo de Producción	Estado de revisión	
Centro de beneficio	1102	<input type="checkbox"/> Suministro prod.	
Clase de objeto	FAB.	Reserva	1702129
Nº de secuencia	0		

Una vez concluida la fabricación del producto se crea la entrega de salida para el cliente, figura 6.

Figura 6. Entrega de salida SAP ERP

Entrega reposición 80000000 Visualizar: Resumen

Contabilizar SM

Entrega salida: 80000000 Fecha documento: 23.01.2009
 Dest.mercancías: C1601 Modaservice II / Av. Los Frutales 329 / LIMA 3 LIMA-ATE

Resumen de posiciones **Picking** Carga Transporte Resumen de status Datos movimientos mercancías

Fe.picking: 23.01.2009 00:0... Stat.gl.picking: C Picking completo
 Núm.almacén: Status glob.WM: No precisa OT p.WM

Pos	Material	Ce.	Alm.	Ctd.entreg.	UM	Ctd.picking	UM	Lote	I...	P	S	FePuestDis	HPue...	Cl.valor.	Denomin.
10	C09MO0021	1201	0001	1.000	UN	1.000	UN					23.01.2009	00:0...		MODULO ELECTR.CONT IL-NT MRS10

Finalmente se Factura el producto. Figura 7 factura en SAP ERP

Figura 7. Factura de ventas en SAP ERP

Factura Ecommerce 9100103154 (ZFEC) Visualizar: Resumen, posiciones de

Finanzas Documentos de facturación

ZFEC Factura Ecommerce 9100103154 Valor neto: 79.00 PEN
 Pagador: 1017210 JOSE DANIEL BARBU FLORES / CALLE 2 NRO. C INT. 6 URB. I...
 Fecha factura: 16.09.2021

Pos.	Denominación	Ctd.facturada	UM	Valor neto	Material	Costes internos	Impt
10	BUJIA CUMMINS MOTOR B GAS PLUS	1.000	UN	12.00	RCU4089085	61.60	2.16
20	CABLE DE BUJIA CUMMINS B GAS	1.000	UN	67.00	RCU3966404	33.61	12.0

El proceso solo por la parte de cotización tenía una duración promedio de 3 días lo cual originaba un malestar en los clientes los cuales optaban por declinar de seguir con el proceso.

Una vez que se tiene la cotización final esta es enviada al departamento de ventas para ser ingresado al sistema este proceso demoraba en promedio un día ya que se tenían que validar los datos del cliente y si la cotización estaba aceptada por gerencia.

3.1.1.1. Problema Principal.

- El proceso de ventas de la empresa automotriz pasa por un excesivo trabajo manual lo que conlleva a una demora general del proceso, originando perdidas en oportunidades de venta.

3.1.1.2. Problemas específicos

- Las cotizaciones para los clientes se crean mediante un Excel lo cual causa que no cumpla con los requerimientos de los clientes, lo que origina un incremento de tiempo para lograr conseguir la cotización que el cliente requiere.
- El trámite burocrático, de aprobación de las cotizaciones aumenta la duración para que la cotización se transforme en un pedido formal de ventas.
- Clientes insatisfechos por la duración del proceso de las ventas.

3.2. Solución

3.2.1 Objetivos

Objetivo General.

Automatizar el proceso de ventas en SAP, para reducir el tiempo de duración del proceso desde la etapa de cotización hasta la Facturación.

Objetivos específicos

- Incrementar el nivel de ventas con el uso de las cotizaciones personalizables.
- Reducir el tiempo del trámite burocrático por medio del manejo estrategias de liberación (Aprobación) en el proceso de ventas.
- Disminuir el número de clientes insatisfechos por los tiempos de duración del proceso de venta.

3.2.2 Alcance

Alcance Funcional

Este proyecto contempló la automatización del proceso de ventas y este incluía:

- Modulo para la generación personalizada de cotizaciones
- Módulo para manejo de estrategias de liberación (Aprobación) a Nivel de gerencias.
- Modulo para la generación de pedido con referencia a la cotización y creación de la orden de producción.
- Modulo para la creación de Facturas.
- Modulo para reportes de clientes y buses
- Modulo para reporte de cotizaciones transformadas en ventas
- Modulo para reporte de eficacia de los vendedores

Alcance Organizacional

El proyecto abarco las Siguiete Áreas:

- Área de ventas
- Área contable
- Área de producción

3.2.3 Etapas y Metodología Usada

3.2.3.1 Etapas de la Implementación del Módulo de cotización y venta

Se tienen las siguientes etapas:

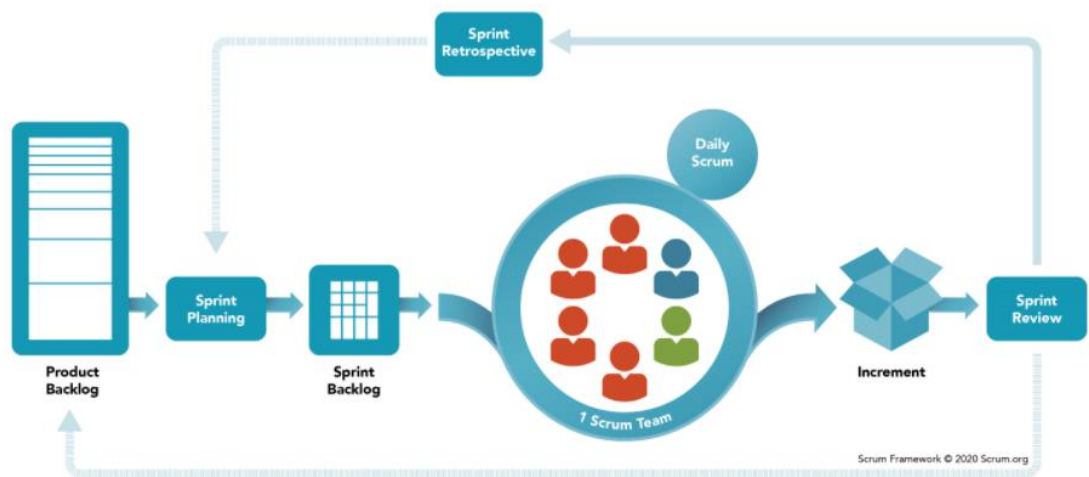
- Análisis y levantamiento de información de proceso de cotización y venta.
- Diseño de la solución
- Elaboración de especificaciones funcionales.
- Implementación de la solución
- Pruebas Unitarias
- Pruebas integrales
- Entrega del Proyecto

3.2.3.2 Metodología Usada

La metodología usada para el proyecto fue la metodología SCRUM.

“SCRUM es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos”. (Schwabe & Sutherland, 2020)

Figura 8. Proceso Scrum (SCRUM ORG, 2021)



3.2.4 Fundamentos Utilizados

Durante las fases de análisis se acordó que el proyecto sería manejado con la metodología SCRUM debido al hecho que el proyecto debía de responder a lo cambiante de los requerimientos que solicitaban los usuarios.

3.2.4.1 Fundamentos para el desarrollo e implementación

“SCRUM es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos”. (Schwabe & Sutherland, 2020)

Valores de SCRUM

- “El Scrum Team se compromete a lograr sus objetivos y a apoyarse mutuamente” (Schwabe & Sutherland, 2020).
- “Su foco principal está en el trabajo del Sprint para lograr el mejor progreso posible hacia estos objetivos” (Schwabe & Sutherland, 2020).
- “El Scrum Team y sus interesados son francos sobre el trabajo y los desafíos” (Schwabe & Sutherland, 2020).
- “Los miembros del Scrum Team se respetan entre sí para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan” (Schwabe & Sutherland, 2020) .

Figura 9. Valores de Scrum (Schwabe & Sutherland, 2020)



ROLES SCRUM:

Product Owner: “Es responsable de maximizar el valor del producto resultante del trabajo del Scrum Team. La forma en que esto se hace puede variar ampliamente entre organizaciones, Scrum Teams e individuos” (Schwabe & Sutherland, 2020).

Scrum Master: “Es responsable de establecer Scrum como se define en la Guía de Scrum. Lo hace ayudando a todos a comprender la teoría y la práctica de Scrum, tanto dentro del Scrum Team como de la organización” (Schwabe & Sutherland, 2020).

Developers: “Las personas del Scrum Team que se comprometen a crear cualquier aspecto de un Increment utilizable en cada Sprint son Developers” (Schwabe & Sutherland, 2020).

EVENTOS SCRUM

El Sprint “Los Sprints son el corazón de Scrum, donde las ideas se convierten en valor. Son eventos de duración fija de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior” (Schwabe & Sutherland, 2020).

Sprint Planning “inicia el Sprint estableciendo el trabajo que se realizará para el Sprint. Este plan resultante es creado por el trabajo colaborativo de todo el equipo de Scrum” (Schwabe & Sutherland, 2020).

Daily Scrum “Es inspeccionar el progreso hacia el Objetivo Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el próximo trabajo planeado” (Schwabe & Sutherland, 2020).

En el proyecto al inicio del día con una duración de no más de 15 minutos se preguntaba al miembro del equipo, como les fue con su trabajo el día de ayer que problemas tuvieron y que trabajo realizaran hoy.

Sprint Review “es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El equipo de Scrum presenta los resultados de su trabajo a las partes interesadas clave y se discute el progreso hacia el Objetivo de Producto” (Schwabe & Sutherland, 2020).

Sprint Retrospective “El propósito de la retrospectiva Sprint es planificar formas de aumentar la calidad y la eficacia” (Schwabe & Sutherland, 2020).

Artefactos SCRUM

Product Backlog “es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente de trabajo emprendida por el equipo Scrum” (SCRUM.ORG, 2020).

- **Historias:** características específicas que el usuario necesita, generalmente se desarrolla dentro del sprint.
- **Tareas:** actividades que el squad necesita realizar.

Sprint Backlog “Se compone del objetivo sprint (por qué), el conjunto de elementos de trabajo pendiente de producto seleccionados para el Sprint (qué), así como un plan accionable para entregar el incremento (cómo)” (Schwabe & Sutherland, 2020).

Incremento “Un Incremento es un peldaño concreto hacia el Objetivo del Producto. Cada Incremento suma a todos los Incrementos anteriores y se verifica minuciosamente, lo que garantiza que todos los Incrementos funcionen juntos. Para proporcionar valor, el Incremento debe ser utilizable” (Schwabe & Sutherland, 2020).

3.2.5. Implementación de las Áreas de procesos y buenas practicas

3.2.5.1. Gestión

Las tareas que se desarrollaron a nivel de gestión usando la metodología SCRUM fueron:

- **Formación del equipo del proyecto:** El equipo del proyecto estuvo conformado por 5 consultores, 2 consultores funcionales y 2 consultores técnicos
- **Asignación de los roles al equipo del proyecto:** los roles asignados fueron los siguientes.

Product Owner: 2 Consultores funcionales

Developers: 2 Consultores técnicos

Scrum Master: Cristian Sarazu Cotrina

- **Elaboración de las Historias de usuario:** Las historias de usuario desarrollados en el proyecto fueron los siguientes:

Tabla 4 Historias de usuario

Historias	Solicitante	requerimiento	Razón
H001	Vendedor	Crear una cotización de manera personalizada de acuerdo al requerimiento del cliente	Brindar cotizaciones realistas y personalizadas a los Clientes

H002	Gerencia	Aprobar las cotizaciones realizadas por los vendedores	Debido a que una cotización incluye los descuentos y precios debe ser validado por gerencia antes de ser procesada como venta
H003	Vendedor	Generar el pedido de venta con referencia a la cotización	se requiere que una vez aceptada la cotización esta pueda convertirse en un pedido de manera automática y este pueda ser atendido
H004	Operario	Generar la orden de Fabricación	se requiere que, una vez creado el pedido de ventas, el operario pueda generar una orden de fabricación, para que los obreros inicien la fabricación del bus
H005	Vendedor	Generar Factura con referencia al pedido	Se requiere que una vez fabricado el Bus se pueda emitir la factura referente al pedido y cotización
H006	Gerencia	Reporte de ventas por Vendedor	Se requiere medir el grado de eficacia de los vendedores
H007	Gerencia	Reporte de cotizaciones terminadas en venta	Se requiere medir el grado de eficacia de los vendedores

3.2.5.2. Análisis de requerimientos: Cada mes se realizaba la revisión del Sprint y planificación para el siguiente Sprint. Los Sprint realizados fueron los siguientes.

Tabla 5 Sprint, historias de usuario y tareas realizadas

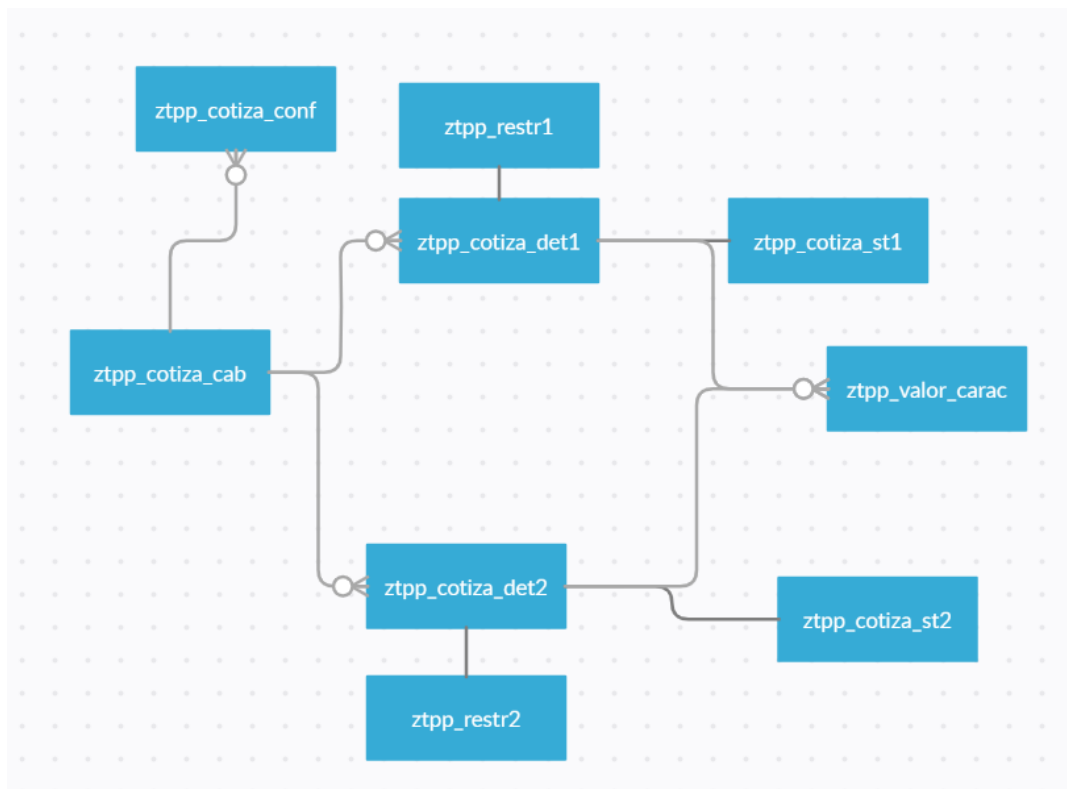
Sprint	Historia	Tarea
S1	H001. Crear una cotización de manera personalizada de acuerdo al requerimiento del cliente	T1. Diseñar el modelo de la base de datos T2. Crear tablas para el almacenamiento de datos maestros de las cotizaciones T3. Cargar los datos maestros, de piezas y modelos T4. Definir el diseño del formato de la cotización T5. Crear programa para la generación de la cotización
S2	H002. Aprobar las cotizaciones realizadas por los vendedores H003. Generar el pedido de venta con referencia a la cotización	T6. Definir las estrategias de liberación (aprobación) T7. Crear workflow de envío de correo para las liberaciones (Aprobaciones) T8. Crear programa de generación automática de pedido de ventas con referencia la cotización
S3	H004. Crear una orden de Fabricación automática H005. Crear Factura con referencia al pedido	T9. Análisis de la creación de ordenes de fabricación T10. Análisis del proceso de facturación T11. Crear programa para la creación de orden de fabricación automática T12. Crear programa para la creación automática de factura con referencia al pedido

3.2.5.3. Diseño.

Todo el proyecto se realizó sobre el ERP SAP. La figura 5 y 6 muestran la cotizacion y características estandar que maneja el SAP.

- El diseño del modelo de la base de dato se muestra en la figura 10 las tablas creadas y las relaciones entre estas.

Figura 10. Diseño de la base de datos



ZTPP_COTIZA_CAB: Tabla de cabecera de las cotizaciones

ZTPP_COTIZA_CONF: Tabla de configuración de la cotización contiene la estructura de cada cotización.

ZTPP_COTIZA_DET Y ZTPP_COTIZA_DET2: Tablas del detalle de la cotización y características que contiene cada cotización.

ZTPP_COTIZA_ST y ZTPP_COTIZA_ST2: Tablas del estatus de las características de las cotizaciones.

ZTPP_RESTR y ZTPP_RESTR: Tablas que contiene las restricciones entre combinaciones de características.

ZTPP_VALOR_CARAC: Tabla de valores posibles por característica

- El diseño de la interfaz del módulo de cotización, se muestra en la figura 11, como se ve en la cotización presenta los botones para aceptarlo o rechazarlo esta opción solo puede ser visualizada por la gerencia que son los encargados de realizar la aceptación o rechazo de la cotización, cuando una cotización es aceptada internamente se crea el pedido de ventas, y la orden de Fabricación. El programa que hace posible la creación de la cotización se encuentra en el ANEXO-01, este programa fue realizado en el lenguaje ABAP.

Figura 11 Modelo de cotización

Imprimir Cotización Confirmar Crear con referencia Rechazar cotización Log de modificaciones

Cabecera		Precio	
N° Cotización	3000000007	Vigente del	31.01.2017 al 25.02.2017
Tipo de Cotización	2 Nacional	Liberado	
Modelos Disponibles	0000000103		
Solicitante	99999	EMPRESA DE TRANSPORTES LUIS BANCH.	
Ciente Final	99999	EMPRESA DE TRANSPORTES LUIS BANCH.	
País	PE	Perú	
Condición de Pago	F030	Factura 30 Dias	
Vendedor	1838	MELLADO VIVANCO LUIS ANTONIO	
Tipo Vta	BUS		
Material	T31UR00018		
Modelo Bus/Carrocería	TITAN		
Tipo Carrocería	URBANO		

Resumen Chasis	
Marca Chasis	MERCEDES BENZ
Modelo Chasis	OF 1721/59
Combustible	GNV
Motor	CUMMINS
Modelo Motor	BGE 230
Formula Rodante	4X2

Cantidad	10	Plazo de Entrega	45 dias
Año de Fab. chasis	2017	Año modelo	2017
Año de Fab. carrocería	2017		
Tipo de Placa	VERDE		
Modelo Unidad	MTUG		

Longitud	12000 mm	Altura	3160 mm	Voladizo DEL	2465 mm
Ancho	2560 mm	D.E.E.	5950 mm	Voladizo POST	3585 mm

Detalle		Ref.			
Título	Subtítulo	Característica	Subcaracterística	Valor	St.
CHASIS	RETARDADOR DE FRENO			NO	
	NUMERO DE CILINDROS GAS			9	
CARROCERIA	CANTIDAD DE ASIENTOS			48	
	ASIENTO PILOTO			GRAMMER	
	ASIENTO COBRADOR			NO	
SISTEMAS INTERNOS	FORRADO DE PISO			PLANCHA ESTRIADA 100%	
	TAPASOL DE CONDUCTOR			SI	
SISTEMAS EXTERNOS	ESPEJOS RETROVISORES			ESPEJO NUEVO	
CLIMATIZACION	RENOVADORES DE AIRE			NO	
	CLARABOYAS			3	
	DESEMPAÑADOR			AIRE CALIENTE	
SISTEMA ELECTRONICO	RADIO CONSOLA			SI	
	LETRERO DE RUTA FRONTAL			SI	
	LETRERO DE RUTA LATERAL ...			NO	
	LETRERO DE RUTA LATERAL ...			NO	
	LETRERO DE RUTA INTERIOR			NO	
	INDICADOR DE VELOCIDAD			SI	
PUERTAS Y VENTANAS	VIDRIOS LATERALES			CORREDIZOS	
	VIDRIO VIGIA			SI	

Figura 12. Transacción para la creación de factura con referencia a la cotización o pedido

Reporte de cotización

Crear Factura

Cotizacion	Tipo de Venta	Material	ModBus/Car	Tipo Carrocería	Pedido	Modelo Chasis	Marca Chasis	Modelo Unidad	Cantidad	Cotizac.	Solicitante	Nombre
3000000001	BUS	T31UR00016	TITAN	URBANO	1015931	17.210 OD	VOLKSWAGEN	VTUG	10	1015931	TRANSPORTES Y SERVICIOS SANTA CRUZ	
3000000002	BUS	T31UR00016	TITAN	URBANO	1006856	17.210 OD	VOLKSWAGEN	VTUG	30	1006856	SAN FELIPE EXPRES S S.A.	
3000000003	BUS	T31UR00015	TITAN	CORREDOR COMPLEMENTARIO	1009807	OF 1721/59	MERCEDES BENZ	MTUG	10	1009807	EMP DE SERV DE TRANSP STA CATALINA S A	
3000000004	BUS	T31UR00016	TITAN	URBANO	1006435	17.210 OD	VOLKSWAGEN	VTUG	40	1006435	EMPRESA DE TRANSPORTES SOL DE ORO S.A.C.	
3000000005	BUS	T31UR00012	TITAN	CORREDOR COMPLEMENTARIO	1012487	17.210 OD	VOLKSWAGEN	VTUG	50	1012487	EMPRESA DE TRANSPORTES Y SERVICIOS VIRGEN DE LA PUERTA S.	
3000000006	BUS	T31UR00016	TITAN	URBANO	99999	17.210 OD	VOLKSWAGEN	VTUG	1	99999	GRUPO AGORA S.A.C.	
3000000007	BUS	T31UR00018	TITAN	URBANO	99999	OF 1721/59	MERCEDES BENZ	MTUG	10	99999	EMPRESA DE TRANSPORTES LUIS BANCHERO ROSSI S.A.	
3000000008	BUS	T31UR00016	TITAN	URBANO	99999	17.210 OD	VOLKSWAGEN	VTUG	10	99999	EMPRESA DE TRANSPORTES Y SERVICIOS LOS OLIVOS S.A.	
3000000009	BUS	T31UR00016	TITAN	URBANO	1006712	17.210 OD	VOLKSWAGEN	VTUG	10	1006712	EMPRESA DE TRANSP Y SERVICIOS EL PORVENIR S.A.	
3000000010	BUS	T31UR00018	TITAN	URBANO	99999	OF 1721/59	MERCEDES BENZ	MTUG	1	99999	PAULA RAMOS SUAZO	
3000000011	BUS	T31UR00016	TITAN	URBANO	99999	17.210 OD	VOLKSWAGEN	VTUG	1	99999	MGM TRANSPORTE Y CARGA SAC	
3000000012	BUS	T31UR00018	TITAN	URBANO	1000584	OF 1721/59	MERCEDES BENZ	MTUG	20	1000584	EMPRESA DE TRANSPORTES Y SERVICIOS SALVADOR S.A.C.	
3000000013	BUS	T30UR00003	APOLO	URBANO	1009482	10-150 EOD	VOLKSWAGEN	VAUD	10	1009482	LUIS MARAVI ARIAS	
3000000014	BUS	T30UR00003	APOLO	URBANO	99999	10-150 EOD	VOLKSWAGEN	VAUD	10	99999	E.T.VICTOR RAUL HAYA DE LA TORRE S.A.	
3000000015	BUS	T31UR00018	TITAN	URBANO	1000584	OF 1721/59	MERCEDES BENZ	MTUG	20	1000584	EMPRESA DE TRANSPORTES Y SERVICIOS SALVADOR S.A.C.	

3.2.5.4. Despliegue.

El despliegue del proyecto se realizó en julio del 2019, donde los usuarios (vendedores) se les dio acceso a las transacciones (módulo de cotización), cada vendedor salía a buscar clientes potenciales con sus laptops, teniendo como resultado la primera semana la elaboración de más de 40 cotizaciones para los clientes debido a lo intuitivo y flexible que era la creación de la cotización y presentación al cliente.

En cuanto a las demás actividades del proceso, las aprobaciones por parte de gerencia igualmente eran más rápidos ya que al gerente le llegaba un correo con el número de cotización que debía de aprobar y este ingresaba a la cotización indicada y de acuerdo a su criterio aprobaba la cotización o la rechazaba, la aceptación originaba la creación automática del pedido y de la orden de fabricación lo cual agilizaba mucho el flujo del proceso.

Una vez concluida la fabricación del producto los vendedores ingresaban al módulo de creación de factura con referencia a la cotización o pedido lo cual hacía la creación automática de la factura la cual se le enviaba al cliente.

3.3. EVALUACIÓN

3.3.1. Evaluación económica

El costo del proyecto incluye los gastos del equipo del proyecto, en la tabla 7 se detalla los gastos incurridos en el proyecto.

Tabla 6. Costo del proyecto

Equipo del proyecto	Cantidad	Costo x mes	Duración	Costo total
Scrum Master	1	12000	3	S/ 36,000.00
Consultores Funcionales	2	12000	3	S/.72,000.00
Consultores técnicos	2	10000	2	S/.40,000.00
			Total	S/.148,000.00

3.3.2. Beneficios Obtenidos

Los beneficios obtenidos en el proyecto fueron los siguientes:

- ✓ Reducción del tiempo en la creación de las cotizaciones

Tabla 7. Tabla de beneficio en la duración de la creación de las cotizaciones (Fuente Empresa automotriz)

Estado	tiempo de creación de una cotización
Antes del proyecto	3 días
Después del proyecto	10 minutos

- ✓ Automatización del proceso completo de las ventas, todo el flujo del proceso se redujo a 2 transacciones las cuales tenían por completo toda la funcionalidad de la creación de cotización pedidos, ordenes de fabricación m y facturación.
- ✓ Incremento en las ventas del 232% respecto a los 3 meses anteriores a la implementación del proyecto, ver la tabla 8.

Tabla 8. Tabla de beneficio económico obtenido del proyecto (Fuente Empresa automotriz)

ventas de abril a junio del año 2019	ventas de Julio a setiembre del año 2019(Etapa posterior a la implementación del proyecto)	Incremento en ventas
19,698,812.76	45,865,159.62	232%

CAPITULO IV. REFLEXIONE CRITICA DE LA EXPERIENCIA

El autor del presente informe participo del proyecto de automatización del proceso de ventas en SAP desde el inicio hasta el fin del proyecto con el rol de jefe de proyectos (Scrum Master) para una empresa automotriz.

La participación directa y activa en todas las etapas del proyecto ayudo a conocer el entorno en el que se implementa un proyecto los riesgos, las oportunidades las actividades correctivas, preventivas que uno puede planear como respuesta a los riesgos que siempre tienen los proyectos, además que ayudo a conocer más sobre SAP ERP sus procesos y las mejoras que se le pueden hacer para maximizar el beneficio que este trae al ser un ERP.

El usar la metodología Scrum para dirigir el proyecto que al inicio era muy cambiante porque no se tenía claro el alcance total proyecto, ayudo ya que se acopló a lo cambiante de este y respondió a las expectativas esperadas, debido a que por cada sprint hacia la entrega de un producto funcional lo cual ayudaba a saber y dimensionar los requisitos siguientes para al final tener el producto completo.

CAPITULO V. CONCLUSIONES Y RECOMENDACIONES

En este capítulo detallamos lo más relevante del proyecto que está ligado al cumplimiento de los objetivos principales, la obtención de las lecciones aprendidas y las recomendaciones para posteriores mejoras.

5.1. Conclusiones

La automatización del proceso de ventas en SAP ERP fue exitosamente implementada trayendo consigo grandes beneficios para a la empresa.

- ✓ Se incrementaron las ventas en un 232% en los 3 meses posteriores a la implementación del proyecto.
- ✓ Se redujo significativamente el tiempo que tomaba la creación de las cotizaciones.
- ✓ Al ser más rápido el tiempo en que el cliente tenía la cotización aumentaba su interés en la compra del producto, y no desistía de este como hacía antes.
- ✓ EL flujo del proceso de la venta se automatizo en la mayor parte de las actividades lo cual redujo el tiempo de duración del proceso.

Las conclusiones son evidenciadas en el capítulo 3.3.2. Beneficios Obtenidos.

5.2. Recomendaciones.

Se recomienda siempre tener iniciativa en buscar mejoras que se le pueda hacer a SAP ERP ya que al ser un ERP las empresas piensan que todo el beneficio que se puede obtener ya está dado por sus transacciones estándar, pero siempre hay cosas que se puede mejorar a medida de la empresa, lo cual puede ocasionar un aumento competitivo significativo con respecto a la competencia.

Usar la metodología Scrum para manejar proyectos cuyos requerimientos son cambiantes o no se tiene dimensionado el alcance del producto final, si no que a medida que se obtiene un producto funcional se le agrega valor.

5.3. Fuentes de Información.

Schwabe, K., & Sutherland, J. (2020). *La Guía Scrum*. Obtenido de Scrum org:
<https://scrumguides.org/download.html>

SCRUM ORG. (2021). *SCRUM ORG*. Obtenido de
<https://www.scrum.org/resources/what-is-scrum>.

SCRUM.ORG. (2020). *Guia Scrum*. Obtenido de
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Soluciones Empresariales SAP EIRL. (s.f.). <https://www.sesap-consulting.com>. Obtenido de <https://www.sesap-consulting.com/quienes-somos>

5.4. Glosario

SAP ERP: es un software de planificación de recursos empresariales desarrollado por la empresa alemana SAP SE

Estrategia de liberación: Lógica para la aprobación de un documento comercial.

Transacción: Es la ejecución de un programa en SAP ERP que abre una interfaz que cumple una funcionalidad.

ABAP: Lenguaje de programación usado para crear mejoras y nuevos proyectos en SAP ERP.

Roll Out: Copia de la implementación SAP ERP de una sucursal de una empresa a otra sucursal que tiene otras restricciones legales u otra forma de organización.

Vendedores de Campo: Vendedores los cuales se desplazan a la ubicación del cliente para presentarles cotizaciones y propuestas para ventas.

SAP FIORI: Tecnología SAP para el desarrollo de aplicaciones móviles o WEB.

ANEXOS

ANEXO-01

Programa ZPPP_COTIZACION_FINAL.

Declaración de variables: ZPPP_COTIZACION_FINAL_TOP -----

```

*&-----*
*& Include          ZPPP_COTIZACION_TOP
*&-----*
*alv data
TABLES: zpp_cotiza_det.

TYPES: BEGIN OF gty_correo ,
        correo TYPE adr6-smtp_addr,
        END OF gty_correo.
DATA: gt_correo TYPE STANDARD TABLE OF gty_correo,
      wa_correo TYPE gty_correo.
DATA: gv_cancelar_accion.
DATA: objeto_carga_imagen TYPE string.
DATA: flag_cot.
DATA: l_flete(10) TYPE c VALUE 'Flete'.
DATA: l_incoterm(10) TYPE c VALUE 'Incoterm'.
DATA: gv_imagen TYPE w3_qname.
DATA: gv_validacion TYPE c.
DATA: dmp200_m_text(70) TYPE c.
DATA: dmp200_m_text_s(70) TYPE c.
DATA: gv_vig_cot TYPE ze_valor.
DATA: gv_plazo_ent TYPE ze_valor.

DATA: gv_aut_precios VALUE 'X'.
DATA: t_tip_placa TYPE STANDARD TABLE OF ztpp_constant WITH HEADER LINE.
TYPES: BEGIN OF ty_datos,
        num_cot          TYPE i,
        campo01(132), "tipo de venta
        campo02(132), "año modelo
        campo03(132), "modelo Bus/Car
        campo04(132), "tipo carroceria
        campo05(132), "version
        campo06(132), "marca chasis
        campo07(132), "modelo chasis
        campo08(132), "combustible
        campo09(132), "motor
        campo10(132), "modelo motor
        campo11(132), "formula rodante
        campo12(132), "titulo
        campo13(132), "subtitulo
        campo14(132), "caracteristica
        campo15(132), "subcaracteristica
        campo16(132), "valor
        campo17(132), "precio
        campo18(132), "moneda
        campo19(132), " mano de obra
        campo20(132), "plantilla por defeccto
        campo21(132), "Descripcion de Configuracion
        flag_cab(1), "plantilla por defeccto
        campo22(132), "secuencia de plantilla
        END OF ty_datos.

TYPES: BEGIN OF ty_m_code,

```

```

        codigo          TYPE ze_pp_cod_clave,
        tip_vta         TYPE ze_pp_tip_vta,
        ano            TYPE ze_pp_ano_mod,
        mod_bus        TYPE ze_pp_mod_bus,
        tip_car        TYPE ze_pp_tip_car,
        mar_cha        TYPE ze_pp_marc_cha,
        mod_cha        TYPE ze_pp_mod_cha,
        cobust         TYPE ze_pp_combust,
        motor          TYPE ze_pp_motor,
        mod_motor      TYPE ze_pp_mod_motor,
        form_rod       TYPE ze_pp_form_rod,
        matnr          TYPE matnr,
        secuen_config  TYPE ze_pp_secuen_config,
        texto          TYPE ze_pp_texto,
    END OF ty_m_code.

```

```

DATA: l_m1(6) TYPE c,
      l_m2(6) TYPE c,
      l_m3(6) TYPE c,
      l_m4(6) TYPE c,
      l_m5(6) TYPE c,
      l_m6(6) TYPE c,
      l_m7(6) TYPE c.

```

```

TYPES: BEGIN OF ty_plantilla.
        INCLUDE STRUCTURE ztpp_claves.
TYPES: secuen_config TYPE ze_pp_secuen_config.
TYPES: texto        TYPE ze_pp_texto.
*TYPES: cost_serv   TYPE ze_pp_cost_serv.
*TYPES: cost_man_ob TYPE ze_pp_cost_man_ob.
*TYPES: pv_netto    TYPE ze_pp_pv_netto.

```

```

TYPES: END OF ty_plantilla.

```

```

DATA: v_tpcambio_aux TYPE ukurs_curr.
DATA: mp200_estado_cotiza(50) TYPE c,
      dmp200_aceptado      TYPE c,
      dmp200_confirmado   TYPE c,
      dmp200_aprobado     TYPE c,
      dmp200_rechazado    TYPE c.

```

```

TYPES: BEGIN OF gty_cotiza_outtab.
        INCLUDE STRUCTURE zpp_cotiza_det.
TYPES: field_style TYPE lvc_t_styl,
      match(2)     TYPE c,
      field_col    TYPE slis_t_specialcol_alv.
TYPES: edita      TYPE c,
      visualizar  TYPE c,
*
      END OF gty_cotiza_outtab.

```

```

TYPES: BEGIN OF ty_clientes,
        kunnr TYPE kunnr,
        name1 TYPE name,
        name2 TYPE name,

```

```

        END OF ty_clientes.

TYPES: BEGIN OF ty_vendedores,
        pernr TYPE pernr_d,
        nchmc TYPE m_premn-nchmc,
        vnamc TYPE m_premn-vnamc,
        END OF ty_vendedores.

DATA: t_vendedores TYPE ty_vendedores OCCURS 0 WITH HEADER LINE.
DATA: t_incoterms TYPE tinct OCCURS 0 WITH HEADER LINE.
DATA: t_t052u TYPE t052u OCCURS 0 WITH HEADER LINE.
DATA: dmpv_nombre(70) TYPE c.
DATA lv_clie(10) TYPE c.

TYPES: BEGIN OF gty_f4data,
        valo TYPE char_value.
TYPES: END OF gty_f4data.

        valor TYPE ze_pp_valor,

        END OF ty_valor.

DATA: t_valor TYPE STANDARD TABLE OF ty_valor.

TYPES: BEGIN OF ty_compara,
*         SECUEN_CONFIG TYPE ZE_PP_SECUEN_CONFIG,
*         ord_tit TYPE ze_pp_ord_tit,
        titulo TYPE ze_pp_titulo,
        ord_subt TYPE ze_pp_ord_subt,
        subtitulo TYPE ze_pp_subtitulo,
        ord_carac TYPE ze_pp_ord_carac,
        caracterica TYPE ze_pp_caracteristica,
        ord_subca TYPE ze_pp_ord_subca,
        subcaracteristica TYPE ze_pp_subcaracteristica,
        valor TYPE ze_pp_valor,
        END OF ty_compara.

.

DATA: grid_cotiza TYPE REF TO cl_gui_alv_grid,
      gt_fieldcatalog_cotiza TYPE lvc_t_fcat,
      gs_layout_cotiza TYPE lvc_s_layo,
*      event_receiver TYPE REF TO lcl_event_receiver,
      gt_exclude_cotiza TYPE ui_functions,
      wa_exclude TYPE ui_func.

DATA: wa_cotiza_outtab TYPE STANDARD TABLE OF gty_cotiza_outtab WITH HEAD
      ER LINE.
DATA: wa_cotiza_outtab_alv TYPE STANDARD TABLE OF gty_cotiza_outtab WITH

```

HEADER LINE.

DATA: answer TYPE c,
gv_error TYPE c.

*-----

DATA: dmp200_macha TYPE ze_pp_marc_cha,
dmp200_mocha TYPE ze_pp_mod_cha,
dmp200_comb TYPE ze_pp_combust,
dmp200_mot TYPE ze_pp_motor,
dmp200_m_mot TYPE ze_pp_mod_motor,
dmp200_f_rod TYPE ze_pp_form_rod.

DATA: dmp200_numcot TYPE ze_pp_num_cot,
dmp200_numcot_ref TYPE ze_pp_num_cot,
dmp200_f_v_cot_d TYPE dats,
dmp200_f_v_cot_h TYPE dats,
dmp200_plazo_ent TYPE ze_pp_plazo_ent,
dmp200_tip_cot TYPE tinc-incol,
dmp200_incoterm TYPE tinc-incol,
dmp200_l_incoterm(70) TYPE c,
dmp200_tip_vta TYPE ze_pp_tip_vta,
dmp200_matnr TYPE matnr,
dmp200_mod_bus TYPE ze_pp_mod_bus,
dmp200_tip_car TYPE ze_pp_tip_car,
dmp200_vers TYPE ze_pp_vers,
dmp200_cant TYPE ze_pp_cantidad VALUE 1,
dmp200_kunnr TYPE kunnr,
dmp200_producto TYPE ze_pp_cod_clave,
dmp200_l_kunnr(70) TYPE c,
dmp200_kunnr2 TYPE ze_pp_kunnr2,
dmp200_l_kunnr2(70) TYPE c,
dmp200_vendedor TYPE ze_pp_vendedor,
dmp200_l_vendedor(70) TYPE c,
dmp200_land1 TYPE land1,
dmp200_l_land1 TYPE text15,
dmp200_zterm TYPE dzterm,
dmp200_l_zterm TYPE text1_052,
dmp200_ano_mod TYPE ze_pp_ano_mod,
dmp200_ano_fab TYPE ze_pp_ano_fab,
dmp200_ano_fab_carr TYPE ze_pp_ano_fab_carr,
dmp200_tip_placa TYPE ze_pp_tip_placa,
dmp200_observ_cotiza TYPE ze_pp_observ_cotiza,

dmp200_moneda TYPE waers,
dmp200_igv TYPE ze_pp_igv,
dmp200_mano_obra TYPE ze_mano_obra,
* dmp200_dcto TYPE ze_pp_dcto,
dmp200_tmp_entrega TYPE ze_tmp_entrega.

DATA: dmp200_precio TYPE ze_pp_precio,
dmp200_costo_total TYPE ze_pp_precio,
dmp200_particip_carroc TYPE ze_pp_precio,
dmp200_particip_chasis TYPE ze_pp_precio,
dmp200_particip_carroc_f TYPE ze_pp_precio,
dmp200_particip_chasis_f TYPE ze_pp_precio,
dmp200_costo_total_marg TYPE ze_pp_precio,
dmp200_dcto TYPE ze_pp_dcto,
dmp200_pv_final TYPE ze_pp_pv_final,

```

dmp200_pv_total          TYPE ze_pp_pv_final,
dmp200_pv_igv           TYPE ze_pp_pv_final,
dmp200_flete            TYPE ze_pp_pv_final,
dmp200_tp_camb          TYPE ze_pp_pv_final.

DATA: dmp200_precio_aux   TYPE ze_pp_precio,
dmp200_costo_total_aux  TYPE ze_pp_precio,
dmp200_dcto_aux        TYPE ze_pp_dcto,
dmp200_margen_aux      TYPE ze_pp_dcto,
dmp200_pv_netto        TYPE ze_pp_pv_final,
dmp200_pv_final_aux    TYPE ze_pp_pv_final,
dmp200_pv_igv_aux      TYPE ze_pp_pv_final,
dmp200_flete_aux       TYPE ze_pp_pv_final,
dmp200_tp_camb_aux     TYPE ze_pp_pv_final,
dmp200_igv_aux         TYPE ze_pp_igv.

DATA: dmp200_long        TYPE ze_pp_longitud,
dmp200_ancho            TYPE ze_pp_ancho,
dmp200_moduni          TYPE ze_modunid,
dmp200_alto            TYPE ze_pp_alto,
dmp200_dee             TYPE ze_pp_dee,
dmp200_vol_del         TYPE ze_pp_voldel,
dmp200_vol_post        TYPE ze_pp_volpos,
dmp200_norma_r66       TYPE ze_pp_norma,
dmp200_chasis          TYPE ze_pp_dcto,
dmp200_chasis_marg     TYPE ze_pp_dcto,
dmp200_margen          TYPE ze_pp_dcto.
*      dmp200_chasism TYPE ze_pp_dcto.

DATA:  dmp200_cost_chasis TYPE ze_pp_precio,
dmp200_cost_chasis2 TYPE ze_pp_precio,
dmp200_cost_carroc TYPE ze_pp_precio,
dmp200_prec_chasis TYPE ze_pp_precio,
dmp200_prec_carroc TYPE ze_pp_precio,
dmp200_prec_netto TYPE ze_pp_precio,
dmp200_prec_bruto TYPE ze_pp_precio,
dmp200_igv1 TYPE ze_pp_precio
.

DATA:  dmp200_prec_netto_f TYPE ze_pp_precio,
dmp200_prec_bruto_f TYPE ze_pp_precio,
dmp200_prec_venta_f TYPE ze_pp_precio,
dmp200_igv_f TYPE ze_pp_precio,
dmp200_dcto_f TYPE ze_pp_precio,
dmp200_flete_f TYPE ze_pp_precio
.

*data: dmp200_aprobado TYPE c.
"IMAGEN
i
"PARA EL ALV1
DATA: gs_layout TYPE lvc_s_layo,
r_contenedor TYPE REF TO cl_gui_custom_container,
r_alv TYPE REF TO cl_gui_alv_grid,
gx_variant2 TYPE disvariant,
gt_exclude2 TYPE ui_functions,
t_sort_alv TYPE .

```

"PARA EL ALVI

DATA: gv_cod_ref2 TYPE ze_pp_cod_ref,
gv_secuen_ref2 TYPE ze_pp_secuen_ref.

*TIP_PLACA ZE_PP_TIP_PLACA

DATA: gv_tip_placa TYPE ze_pp_tip_placa,
gv_mano_obra TYPE ze_pp_cost_serv,
gv_dcto TYPE ze_pp_cost_man_ob,
gv_cost_serv TYPE ze_pp_marg_gananc,
gv_cost_man_ob TYPE ze_pp_flete,
gv_cost_seguridad TYPE ze_pp_flete,
gv_marg_gananc_l TYPE ze_pp_pv_netto,
gv_marg_gananc_h TYPE ze_pp_pv_netto,
gv_marg_ganancha_l TYPE ze_pp_pv_netto,
gv_marg_ganancha_h TYPE ze_pp_pv_netto..

DATA: gv_codigo TYPE ze_pp_cod_clave.

DATA: t_match TYPE ddshretval OCCURS 0 WITH HEADER LINE.

DATA: t_valores TYPE STANDARD TABLE OF ty_valor WITH HEADER LINE.

DATA: gt_claves TYPE STANDARD TABLE OF ztpp_claves WITH HEADER LINE.

DATA : t_name TYPE TABLE OF dynpread,
wls_dyfield TYPE dynpread.

DATA: gwa_plant TYPE ty_plantilla.

DATA: ge_ucomm TYPE c,
gv_ucomm2 TYPE c.. "M:editar C:crear

DATA: dmp300_c_ruc TYPE stcd1,
dmp300_c_dni TYPE stcd1,
dmp300_c_kunnr TYPE kunnr,
dmp300_c_nombre(150) TYPE c,
dmp300_c_contacto(150) TYPE c,
dmp300_c_correo TYPE adr6-smtp_addr,
dmp300_c_telefono TYPE knal-telf1,
dmp300_c_direccion(225) TYPE c.

TYPES: BEGIN OF gty_data,
descr TYPE atbez,
valo TYPE char_value,
name TYPE atnam.

TYPES: cell_tab TYPE lvc_t_styl.

TYPES: END OF gty_data.

DATA: classcharacteristics TYPE STANDARD TABLE OF bapi1003_charact_r WITH
H HEADER LINE,

classcharactervalues TYPE STANDARD TABLE OF bapi1003_char_val_r WI
TH HEADER LINE,

classdescriptions TYPE STANDARD TABLE OF bapi1003_catch_r WITH
HEADER LINE,

classlongtexts TYPE STANDARD TABLE OF bapi1003_longtext_r WI
TH HEADER LINE.

DATA: gt_fieldcat2 TYPE lvc_t_fcat,
r_fieldcat2 TYPE lvc_s_fcat,

```

class2          TYPE klah-class..

DATA :feature_info TYPE STANDARD TABLE OF kldat WITH HEADER LINE,
      objectkeynew TYPE bapil003_key-object.

DATA: gt_data TYPE STANDARD TABLE OF gty_data WITH HEADER LINE.

DATA: gt_datac TYPE STANDARD TABLE OF gty_data WITH HEADER LINE.
DATA: t_newtable TYPE REF TO data,
      t_newtable2 TYPE REF TO data,
      t_newline  TYPE REF TO data.

FIELD-
SYMBOLS: <t_dyntable>          TYPE STANDARD TABLE, " Dynamic internal table
name
          <fs_dyntable>       TYPE any, " Field symbol to create work
area
          <fs fldval>         TYPE any,
          <t_dyntable_mach>    TYPE STANDARD TABLE.

*-----
'
wa_plantilla1 TYPE ztpp_valor1,
wa_plantilla2 TYPE ztpp_valor2,
wa_flag2      TYPE ztpp_cotiza_st2, "ztpp_valor2,
wa_flag1      TYPE ztpp_cotiza_st1, "ztpp_valor1,
wa_config     TYPE ztpp_configura.

DATA: wa_output TYPE ty_plantilla.

DATA: gv_margen_chas_sup TYPE ze_pp_dcto,
      gv_margen_chas_inf TYPE ze_pp_dcto.

DATA: dmp200_mot_rech(30),
      dmp200_motivo_rech(150),
      dmp200_obs_rech(150).

DATA: flag_mod.

DATA: gv_acceso_all TYPE c.

```

pdf

DATA: gs_otfdata TYPE ssfcrescl.

dmp201_fcat TYPE lvc_t_fcat,
dmp201_alv1 TYPE REF TO cl_gui_alv_grid,
dmp201_cont TYPE REF TO cl_gui_custom_container,
dmp201_vari TYPE disvariant,
dmp201_excl TYPE ui_functions,
dmp201_layo TYPE lvc_s_layo,
dmp201_f401 TYPE lvc_t_f4 WITH HEADER LINE,
is_stable TYPE lvc_s_stbl.

DATA: gv_valida_archivos TYPE c.

Declaración de Pantalla: ZPPP_COTIZACION_FINAL_SEL -----

```
*&-----*  
*& Include          ZPPP_COTIZACION_SEL  
*&-----*
```

```
*SELECTION-SCREEN BEGIN OF BLOCK bloque2 WITH FRAME TITLE text-200.  
SELECTION-SCREEN BEGIN OF SCREEN 000 AS WINDOW TITLE text-200.
```

SELECTION-SCREEN: FUNCTION KEY 1.

SELECTION-SCREEN END OF SCREEN 000.

```
*SELECTION-SCREEN END OF BLOCK bloque2.
```

```
SELECTION-SCREEN BEGIN OF BLOCK bloque1 WITH FRAME TITLE text-100.
```

```
PARAMETERS: p_tvta TYPE ze_pp_tip_vta NO-DISPLAY,  
             p_ano  TYPE gjahr NO-DISPLAY,  
             p_mbus TYPE ze_pp_mod_bus NO-DISPLAY,  
             p_tcar TYPE ze_pp_tip_car NO-DISPLAY,  
             p_vers TYPE ze_pp_vers NO-DISPLAY,  
             p_macha TYPE ze_pp_marc_cha NO-DISPLAY,  
             p_mocha TYPE ze_pp_mod_cha NO-DISPLAY,  
             p_comb TYPE ze_pp_combust NO-DISPLAY,  
             p_mot  TYPE ze_pp_motor NO-DISPLAY,
```



```

p_m_mot TYPE ze_pp_mod_motor NO-DISPLAY,
p_f_rod TYPE ze_pp_form_rod NO-DISPLAY.

```

```

SELECTION-SCREEN END OF BLOCK bloquel.

```

Declaración de clases: ZPPP_COTIZACION_FINAL_SEL -----

```

*&-----*
*& Include          ZPPP_COTIZACION_CLA
*&-----*

```

```

CLASS lcl_event_handler DEFINITION.

```

```

  PUBLIC SECTION .

```

```

  METHODS:

```

```

    handle_user_command
      FOR EVENT user_command OF cl_gui_alv_grid
      IMPORTING e_ucomm,

```

```

    handle_toolbar
      FOR EVENT toolbar OF cl_gui_alv_grid
      IMPORTING e_object
              e_interactive,

```

```

*      handle_top_of_page FOR EVENT top_of_page OF cl_gui_alv_grid IMPORT
ING e_dyndoc_id,

```

```

ENDCLASS. "lcl_event_handler DEFINITION

```

```

CLASS lcl_event_handler IMPLEMENTATION .

```

```

* METHOD handle_user_command.

```

```

** break dennisu.

```

```

* CASE e_ucomm.

```

```

** WHEN 'PAIS'.

```

```

** PERFORM resumen.

```

```

* WHEN 'DESCARGA'.

```

```

*      PERFORM descargar_xml.
*      WHEN 'SOLES'.
*      PERFORM catalogo2.
*      CALL METHOD r_alv2->set_frontend_fieldcatalog
*      EXPORTING
*          it_fieldcatalog = it_catalogo2[].
*      CALL METHOD r_alv2->refresh_table_display.
*      WHEN 'DOLAR'.
*      PERFORM catalogod.
= it_catalogo2[].
*      CALL METHOD r_alv2->refresh_table_display.
*
*
*      ENDCASE.
*
**      OTHERS      = 2.
*
*      ENDMETHOD. "handle_user_command

```

```

METHOD handle_double_clickgrid.

```

```

    DATA: wa_output TYPE ty_plantilla.
    DATA: lt_config TYPE STANDARD TABLE OF ztpp_config.
    DATA: wa_config TYPE ztpp_config.
    DATA: flag_editable(1) TYPE c.
    DATA: lwa_cotiza_outtab TYPE gty_cotiza_outtab.
    REFRESH: gt_cotiza_outtab_alv,gt_cotiza_outtab.
    DATA: lw_longitud TYPE ztpp_longitud.

```

```

    REFRESH gt_cotiza_outtab.

```

```

    READ TABLE gt_plantillas_st INTO wa_output INDEX e_row-index.

```

```

    IF sy-subrc = 0.

```

```

        p_tvta = wa_output-tip_vta.

```

```

        p_ano  = wa_output-ano.

```

```

        p_mbus = wa_output-mod_bus.

```

```

        p_tcar = wa_output-tip_car.

```

```

        p_vers = wa_output-vers.

```

```

        p_macha = wa_output-mar_cha.

```

```

p_mocha = wa_output-mod_cha.
p_comb  = wa_output-cobust.
p_mot   = wa_output-motor.
p_m_mot = wa_output-mod_motor.
p_f_rod = wa_output-form_rod.
PERFORM cargar_constantes.
*      PERFORM consultar_plantillas.

*****INI COMENT*****
*****

*      gv_codigo = wa_output-codigo.
*      SELECT * FROM ztpp_config INTO TABLE lt_config WHERE
*      codigo = wa_output-codigo AND
*      secuen_config = wa_output-secuen_config.
*      IF sy-subrc = 0.
**dmp200_numcot
*      GV_COD_REF2      = wa_output-codigo.
*      GV_SECUEN_REF2 = wa_output-secuen_config.
*      dmp200_tip_vta = wa_output-tip_vta.
*      dmp200_matnr   = wa_output-matnr.
*      dmp200_mod_bus = wa_output-mod_bus.
*      dmp200_tip_car = wa_output-tip_car.
*      dmp200_vers    = wa_output-vers.
*      CLEAR lw_longitud .
*
*      select SINGLE * into lw_longitud from ZTPP_LONGITUD
*      where matnr = dmp200_matnr .
*
*
*      dmp200_long      = lw_longitud-LONGITUD.
*      dmp200_ancho    = lw_longitud-ANCHO.
*      dmp200_alto     = lw_longitud-ALTO.
*      dmp200_dee      = lw_longitud-DEE.
*      dmp200_vol_del  = lw_longitud-VOL_DEL.
*      dmp200_vol_post = lw_longitud-VOL_POST.
*      dmp200_norma_r66 = lw_longitud-NORMA_R66.
*
**dmp200_cant      = wa

```

```

**dmp200_kunnr
**dmp200_kunnr2
**dmp200_land1
**dmp200_ano_mod = wa_output-
**dmp200_ano_fab
*
*      LOOP AT lt_config INTO wa_config.
*      CLEAR lwa_cotiza_outtab.
*      MOVE-CORRESPONDING wa_config TO lwa_cotiza_outtab.
*      lwa_cotiza_outtab-st_fila = ''.
*      lwa_cotiza_outtab-icon = '@5B@'.
*      APPEND lwa_cotiza_outtab TO gt_cotiza_outtab.
**      PERFORM valor_editable USING wa_config-titulo wa_config-
subtitulo wa_config-caracterica wa_config-subcaracteristica
**      CHANGING flag_editable.
**      @5B@
**      @04@
**      @11@
*      ENDLOOP.
*
*
*
*
*
*
*      CALL SCREEN 200.
*      ENDIF.
*
**      PERFORM mostrar_detalle.
*****END COMENT*****
*****
      ENDIF.

      ENDMETHOD. "handle_top_of_page

METHOD handle_f4_help.
      DATA: r_f4data TYPE LINE OF gt_f4data.
      REFRESH gt_f4data.
      CLEAR gt_f4data.

```

```

*   READ TABLE ti_asignacion ASSIGNING <fv_data> INDEX es_row_no-
row_id.
*   IF sy-subrc = 0.
*
*   ENDIF.
* CALL METHOD grid_cotiza->check_changed_data.
   PERFORM handle_mach3 USING e_fieldname es_row_no.
   er_event_data->m_event_handled = 'X'.
   ENDMETHOD. "HANDLE_F4_HELP
* METHOD handle_top_of_page.
*   PERFORM event_top_of_page USING o_dyndoc_id.
* ENDMETHOD. "handle_top_of_page

```

```

"HANDLE_DATA_CHANGED2
"handle_double_clickgrid

```

```

METHOD handle_toolbar.
*   DATA: ls_toolbar TYPE stb_button.
**Agregar nivel
*   CLEAR ls_toolbar.
*   MOVE 'AGREGAR_NIVEL' TO ls_toolbar-function.
*   MOVE icon_insert_row TO ls_toolbar-icon.
*   MOVE 'Agregar nivel'(112) TO ls_toolbar-text.
*   MOVE 'Agregar nivel'(112) TO ls_toolbar-quickinfo.
*   MOVE ' ' TO ls_toolbar-disabled.
**   IF g_accion NE 'CONSULTAR'.
*   APPEND ls_toolbar TO e_object->mt_toolbar.
**   ENDF.
**Agregar subnivel
*   CLEAR ls_toolbar.
*   MOVE 'AGREGAR_SUBNIVEL' TO ls_toolbar-function.
*   MOVE icon_retrieve TO ls_toolbar-icon.
*   MOVE 'Agregar subnivel'(112) TO ls_toolbar-text.
*   MOVE 'Agregar subnivel'(112) TO ls_toolbar-quickinfo.
*   MOVE ' ' TO ls_toolbar-disabled.
**   IF g_accion NE 'CONSULTAR'.

```

```

*   APPEND ls_toolbar TO e_object->mt_toolbar.
**   ENDIF.
**Borrar nivel
*   CLEAR ls_toolbar.
*   MOVE 'BORRAR_NIVEL' TO ls_toolbar-function.
*   MOVE icon_delete_row TO ls_toolbar-icon.
*   MOVE 'Borrar nivel'(112) TO ls_toolbar-text.
*   MOVE 'Borrar nivel'(112) TO ls_toolbar-quickinfo.
*   MOVE ' ' TO ls_toolbar-disabled.
**   IF g_accion NE 'CONSULTAR'.
*   APPEND ls_toolbar TO e_object->mt_toolbar.
*   ENDIF.

```

ENDMETHOD.

METHOD handle_user_command.

```

FIELD-SYMBOLS: <fs_cells>          TYPE LINE OF lt_cells,
                <fs_tit>           TYPE LINE OF s_tit,
                <fs_subtit>        TYPE LINE OF s_subtit,
                <fs_carac>         TYPE LINE OF s_carac,
                <fs_carac1>        TYPE LINE OF s_scarac,
                <fs_cotiza_outtab> TYPE LINE OF gt_cotiza_outtab..

```

DATA: dmpv_tiplinea TYPE i.

DATA: gt_cotiza_outtab_aux TYPE STANDARD TABLE OF gty_cotiza_outtab.

DATA: wa_cotiza_outtab_aux TYPE LINE OF gt_cotiza_outtab,
wa_cotiza_outtab_aux2 TYPE LINE OF gt_cotiza_outtab.

DATA: is_stable TYPE lvc_s_stbl,
w_cell TYPE LINE OF lt_cells,
v_add TYPE i,
v_posi TYPE i.

DATA: dmpv_lines TYPE i.

DATA: cont TYPE i.

is_stable-row = 'X'.

is_stable-col = 'X'.

```

CALL METHOD grid_cotiza->check_changed_data.

REFRESH :s_tit,
s_subtit,
s_carac,
s_scarac.

CLEAR: lv_lines.
DESCRIBE TABLE gt_cotiza_outtab LINES lv_lines.

CLEAR v_posi.
CASE e_ucomm.
  WHEN 'AGREGAR_NIVEL'.
    CALL METHOD grid_cotiza->get_selected_cells
      IMPORTING
        et_cell = lt_cells.

    READ TABLE lt_cells INTO w_cell INDEX 1.
    IF sy-subrc = 0.
      READ TABLE gt_cotiza_outtab_alv INTO wa_cotiza_outtab_alv INDEX
w_cell-row_id.
      IF sy-subrc = 0 AND wa_cotiza_outtab_alv-st_fila NE '2'.
        CASE w_cell-col_id.
          WHEN 'TITULO'.
            IF wa_cotiza_outtab_alv-titulo IS NOT INITIAL.
              CALL SELECTION-SCREEN 001 STARTING AT 20 5.
              IF s_tit[] IS NOT INITIAL.
                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
                LOOP AT s_tit ASSIGNING <fs_tit>.

*           READ TABLE gt_cotiza_outtab INTO wa_ultimo INDEX L
V_LINES.
*           if sy-subrc = 0 and wa_ultimo-
titulo = wa_cotiza_outtab_aux-titulo.
*
*           ENDIF.

```

```

                                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = <fs_tit>-low .
                                IF sy-subrc <> 0.
                                    CLEAR cont.
                                LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_au
x.
                                    IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit OR wa_cotiza_outtab_aux-
titulo = wa_cotiza_outtab-titulo .
                                        wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                                        APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
                                    ELSE.
                                        IF cont = 0.
*
                                        wa_cotiza_outtab_aux2 = wa_cotiza_outtab_
aux.
                                        CLEAR:wa_cotiza_outtab_aux2-
subtitulo, wa_cotiza_outtab_aux2-caracterica, wa_cotiza_outtab_aux2-
subcaracteristica,
                                        wa_cotiza_outtab_aux2-
subtitulo,wa_cotiza_outtab_aux2-valor.
                                        wa_cotiza_outtab_aux2-titulo = <fs_tit>-
low.
                                        wa_cotiza_outtab_aux2-icon = '@17@'.
                                        wa_cotiza_outtab_aux2-st_fila = '1'.
                                        ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                        APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.
                                        ADD 1 TO cont.
                                    ENDIF.
                                    ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
                                    APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
                                ENDIF.
                                IF sy-tabix = lv_lines AND cont <> 1.
                                    CLEAR:wa_cotiza_outtab_aux2-
subtitulo, wa_cotiza_outtab_aux2-caracterica, wa_cotiza_outtab_aux2-

```



```

subcaracteristica,
                                wa_cotiza_outtab_aux2-
subtitulo,wa_cotiza_outtab_aux2-valor..
                                wa_cotiza_outtab_aux2-titulo = <fs_tit>-low.
                                wa_cotiza_outtab_aux2-icon = '@17@'.
                                wa_cotiza_outtab_aux2-st_filas = '1'.
                                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

                                ENDIF.

                                ENDLOOP.
                                REFRESH: gt_cotiza_outtab.
                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
                                PERFORM procesa_data.

                                ELSE.
                                MESSAGE 'El Titulo ya existe' TYPE 'S' DISPLAY TY
PE 'E'.

                                ENDIF.
                                ENDLOOP.

                                ENDIF.
                                ENDIF.
                                WHEN 'SUBTITULO'.
                                IF wa_cotiza_outtab_alv-subtitulo IS NOT INITIAL.
                                CALL SELECTION-SCREEN 002 STARTING AT 20 5.
                                IF s_subtit[] IS NOT INITIAL.
                                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
                                LOOP AT s_subtit ASSIGNING <fs_subtit>.
                                *
                                CLEAR: lv_lines.
                                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

                                subtitulo = <fs_subtit>-low.
                                IF sy-subrc <> 0.
                                CLEAR cont.
                                LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux.

                                IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit OR ( wa_cotiza_outtab_aux-

```

```

titulo = wa_cotiza_outtab-titulo AND
                                wa_cotiza_outtab_aux-
subtitulo = wa_cotiza_outtab-subtitulo ).
                                wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

ELSE.
    IF cont = 0.

        CLEAR: wa_cotiza_outtab_aux2-
caracterica, wa_cotiza_outtab_aux2-
subcaracteristica, wa_cotiza_outtab_aux2-valor..
                                wa_cotiza_outtab_aux2-
subtitulo = <fs_subtit>-low.
                                wa_cotiza_outtab_aux2-icon = '@17@'.
                                wa_cotiza_outtab_aux2-st_filas = '1'.
                                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_
outtab_aux.

                                ADD 1 TO cont.
ENDIF.
                                ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
                                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

ENDIF.
                                IF sy-tabix = lv_lines AND cont <> 1.
                                    CLEAR: wa_cotiza_outtab_aux2-
subtitulo, wa_cotiza_outtab_aux2-caracterica, wa_cotiza_outtab_aux2-
subcaracteristica, wa_cotiza_outtab_aux2-subtitulo.
                                    wa_cotiza_outtab_aux2-
subtitulo = <fs_subtit>-low.
                                    wa_cotiza_outtab_aux2-icon = '@17@'.
                                    wa_cotiza_outtab_aux2-st_filas = '1'.
                                    ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                    APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

                                    ENDIF.

                                ENDLIST.
                                REFRESH: gt_cotiza_outtab.
                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].

```

```

        PERFORM procesa_data.
    ELSE.
        MESSAGE 'El Subtitulo ya existe' TYPE 'S' DISPLAY
TYPE 'E'.

        ENDIF.
    ENDLOOP.

*      REFRESH: gt_cotiza_outtab.
*      gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
*      PERFORM procesa_data.

    ENDIF.
ENDIF.

WHEN 'CARACTERICA'.
    IF wa_cotiza_outtab_alv-caracterica IS NOT INITIAL.
        CALL SELECTION-SCREEN 003 STARTING AT 20 5.
        IF s_carac[] IS NOT INITIAL.
            READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.

            LOOP AT s_carac ASSIGNING <fs_carac>.
*              CLEAR: lv_lines.
                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

                subtitulo = wa_cotiza_outtab-subtitulo

                caracterica = <fs_carac>-low.
                IF sy-subrc <> 0.
                    CLEAR cont.
                    LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_a
x.

                        IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit OR ( wa_cotiza_outtab_aux-
titulo = wa_cotiza_outtab-titulo AND
                                wa_cotiza_outtab_aux-
subtitulo = wa_cotiza_outtab-subtitulo AND wa_cotiza_outtab_aux-
caracterica = wa_cotiza_outtab-caracterica ).
                                    wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.

                                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

                        ELSE.

```

```

        IF cont = 0.

            CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                wa_cotiza_outtab_aux2-
caracterica = <fs_carac>-low.
                wa_cotiza_outtab_aux2-icon = '@17@'.
                wa_cotiza_outtab_aux2-st_fila = '1'.
                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza
_outtab_aux.

                ADD 1 TO cont.
            ENDIF.
            ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
            APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

        ENDIF.

        IF sy-tabix = lv_lines AND cont <> 1.
            CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica, wa_cotiza_outtab_aux2-valor.
                wa_cotiza_outtab_aux2-
caracterica = <fs_carac>-low.
                wa_cotiza_outtab_aux2-icon = '@17@'.
                wa_cotiza_outtab_aux2-st_fila = '1'.
                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

            ENDIF.
        ENDLOOP.
        REFRESH: gt_cotiza_outtab.
        gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
        PERFORM procesa_data.
    ELSE.
        MESSAGE 'La Caracteristica ya existe' TYPE 'S' DI
SPLAY TYPE 'E'.

        ENDIF.
    ENDLOOP.
*           REFRESH: gt_cotiza_outtab.
*           gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
*           PERFORM procesa_data.
ENDIF.

```

```

ENDIF.

WHEN 'SUBCARACTERISTICA'.
  IF wa_cotiza_outtab_alv-subcaracteristica IS NOT INITIAL.
    CALL SELECTION-SCREEN 004 STARTING AT 20 5.
    IF s_scarac[] IS NOT INITIAL.
      READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
      LOOP AT s_scarac ASSIGNING <fs_carac1>.
*         CLEAR lv_lines.
          READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

          subtitulo = wa_cotiza_outtab-subtitulo

          caracterica = wa_cotiza_outtab-caracterica

          subcaracteristica = <fs_carac1>-low.
          IF sy-subrc <> 0.
            CLEAR cont.
            LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_au
x.
              IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit OR ( wa_cotiza_outtab_aux-
titulo = wa_cotiza_outtab-titulo AND
              wa_cotiza_outtab_aux-
subtitulo = wa_cotiza_outtab-subtitulo AND wa_cotiza_outtab_aux-
caracterica = wa_cotiza_outtab-caracterica AND
              wa_cotiza_outtab_aux-
subcaracteristica = wa_cotiza_outtab-subcaracteristica ).
                wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
              ELSE.
                IF cont = 0.

                  CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                  wa_cotiza_outtab_aux2-
subcaracteristica = <fs_carac1>-low.

```

```

        wa_cotiza_outtab_aux2-icon = '@17@'.
        wa_cotiza_outtab_aux2-st_fila = '1'.
        ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
        APPEND wa_cotiza_outtab_aux2 TO gt_cotiza
_outtab_aux.

        ADD 1 TO cont.
    ENDIF.
    ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
    APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

    ENDIF.

    IF sy-tabix = lv_lines AND cont <> 1.
        CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica, wa_cotiza_outtab_aux2-valor.
        wa_cotiza_outtab_aux2-
subcaracteristica = <fs_caracl>-low.
        wa_cotiza_outtab_aux2-icon = '@17@'.
        wa_cotiza_outtab_aux2-st_fila = '1'.
        ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
        APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

        ENDIF.

    ENDLOOP.
    REFRESH: gt_cotiza_outtab.
    gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
    PERFORM procesa_data.
    ELSE.
        MESSAGE 'La Subcaracteristica ya existe' TYPE 'S'
DISPLAY TYPE 'E'.
    ENDIF.

    ENDLOOP.
*
*
*
    REFRESH: gt_cotiza_outtab.
    gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
    PERFORM procesa_data.
    ENDIF.
ENDIF.

ENDCASE.

```

```

        ENDIF.
    ENDIF.

    WHEN 'AGREGAR_SUBNIVEL'.
        CALL METHOD grid_cotiza->get_selected_cells
            IMPORTING
                et_cell = lt_cells.

        READ TABLE lt_cells INTO w_cell INDEX 1.
        IF sy-subrc = 0.
            READ TABLE gt_cotiza_outtab_alv INTO wa_cotiza_outtab_alv INDEX
w_cell-row_id.
            IF sy-subrc = 0 AND wa_cotiza_outtab_alv-st_fila NE '2' .
                CASE w_cell-col_id.
                    WHEN 'TITULO'.
                        IF wa_cotiza_outtab_alv-titulo IS NOT INITIAL.
                            CALL SELECTION-SCREEN 002 STARTING AT 20 5.
                            IF s_subtit[] IS NOT INITIAL.
                                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
                                LOOP AT s_subtit ASSIGNING <fs_subtit>.
*                                    CLEAR lv_lines.
                                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

                                subtitulo = <fs_subtit>-low.
                                IF sy-subrc <> 0.
                                    CLEAR cont.
                                    LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_au
x.
                                        IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit .
                                            wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                                        APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

                                        ELSE.
                                            IF cont = 0.
                                                CLEAR: wa_cotiza_outtab_aux2-

```

```

caracterica, wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor..
                                wa_cotiza_outtab_aux2-
subtitulo = <fs_subtit>-low.
                                wa_cotiza_outtab_aux2-icon = '@17@'.
                                wa_cotiza_outtab_aux2-st_filas = '1'.
                                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_
outtab_aux.

                                ADD 1 TO cont.
                                ENDIF.
                                ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
                                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.

                                ENDIF.

                                IF sy-tabix = lv_lines AND cont <> 1.
                                CLEAR: wa_cotiza_outtab_aux2-
caracterica, wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                                wa_cotiza_outtab_aux2-
subtitulo = <fs_subtit>-low.
                                wa_cotiza_outtab_aux2-icon = '@17@'.
                                wa_cotiza_outtab_aux2-st_filas = '1'.
                                ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

                                ENDIF.

                                ENDLOOP.
                                REFRESH: gt_cotiza_outtab.
                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
                                PERFORM procesa_data.
                                ELSE.
                                MESSAGE 'El subtitulo ya existe' TYPE 'S' DISPLAY
TYPE 'E'.

                                ENDIF.
                                ENDLOOP.
*                                REFRESH: gt_cotiza_outtab.
*                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
*                                PERFORM procesa_data.

```



```

        ENDIF.
    ENDIF.

    WHEN 'SUBTITULO'.
        IF wa_cotiza_outtab_alv-subtitulo IS NOT INITIAL.
            CALL SELECTION-SCREEN 003 STARTING AT 20 5.
            IF s_carac[] IS NOT INITIAL.
                READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
                LOOP AT s_carac ASSIGNING <fs_carac>.
*                    clear lv_lines.
                    READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

                    subtitulo = wa_cotiza_outtab-subtitulo

                    caracterica = <fs_carac>-low.
                    IF sy-subrc <> 0.
                        CLEAR cont.
                        LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_au
x.
                            IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit .
                                wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                                    APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
                                ELSE.
                                    IF cont = 0.
                                        CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                                        wa_cotiza_outtab_aux2-
caracterica = <fs_carac>-low.
                                        wa_cotiza_outtab_aux2-icon = '@17@'.
                                        wa_cotiza_outtab_aux2-st_fila = '1'.
                                        ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                        APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
_outtab_aux.
                                        ADD 1 TO cont.
                                    ENDIF.
                                    ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
                                    APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou

```

```

ttab_aux.

                                ENDIF.

                                IF sy-tabix = lv_lines AND cont <> 1.
                                    CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                                    wa_cotiza_outtab_aux2-
caracterica = <fs_carac>-low.
                                    wa_cotiza_outtab_aux2-icon = '@17@'.
                                    wa_cotiza_outtab_aux2-st_fila = '1'.
                                    ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                                    APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

                                ENDIF.

                                ENDLOOP.
                                REFRESH: gt_cotiza_outtab.
                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
                                PERFORM procesa_data.
                                ELSE.
                                    MESSAGE 'La caracteristica ya existe' TYPE 'S' D
ISPLAY TYPE 'E'.

                                ENDIF.
                                ENDLOOP.
*                                REFRESH: gt_cotiza_outtab.
*                                gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
*                                PERFORM procesa_data.
                                ENDIF.
                                ENDIF.

                                WHEN 'CARACTERICA'.
                                    IF wa_cotiza_outtab_alv-caracterica IS NOT INITIAL.
                                        CALL SELECTION-SCREEN 004 STARTING AT 20 5.
                                        IF s_scarac[] IS NOT INITIAL.
                                            READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab IND
EX w_cell-row_id.
                                            LOOP AT s_scarac ASSIGNING <fs_cara1>.
*                                                CLEAR lv_lines.
                                                    READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab_a
ux WITH KEY titulo = wa_cotiza_outtab-titulo

                                                    subtitulo = wa_cotiza_outtab-subtitulo

```

```

caracterica = wa_cotiza_outtab-caracterica

subcaracteristica = <fs_caracl>-low.
    IF sy-subrc <> 0.
        CLEAR cont.
        LOOP AT gt_cotiza_outtab INTO wa_cotiza_outtab_au
x.
            IF wa_cotiza_outtab-
ord_tit >= wa_cotiza_outtab_aux-ord_tit .
                wa_cotiza_outtab_aux2 = wa_cotiza_outtab_aux
.
                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
            ELSE.
                IF cont = 0.
                    CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
                    wa_cotiza_outtab_aux2-
subcaracteristica = <fs_caracl>-low.
                    wa_cotiza_outtab_aux2-icon = '@17@'.
                    wa_cotiza_outtab_aux2-st_fila = '1'.
                    ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
                    APPEND wa_cotiza_outtab_aux2 TO gt_cotiza
_outtab_aux.
                    ADD 1 TO cont.
                ENDIF.
                ADD 1 TO wa_cotiza_outtab_aux-ord_tit.
                APPEND wa_cotiza_outtab_aux TO gt_cotiza_ou
ttab_aux.
            ENDIF.

    IF sy-tabix = lv_lines AND cont <> 1.
        CLEAR: wa_cotiza_outtab_aux2-
subcaracteristica,wa_cotiza_outtab_aux2-valor.
        wa_cotiza_outtab_aux2-
subcaracteristica = <fs_caracl>-low.
        wa_cotiza_outtab_aux2-icon = '@17@'.
        wa_cotiza_outtab_aux2-st_fila = '1'.
        ADD 1 TO wa_cotiza_outtab_aux2-ord_tit.
        APPEND wa_cotiza_outtab_aux2 TO gt_cotiza_ou
ttab_aux.

```

```

        ENDIF.

    ENDLOOP.

    REFRESH: gt_cotiza_outtab.
    gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
    PERFORM procesa_data.
ELSE.
    MESSAGE 'La Subcaracteristica ya existe' TYPE 'S'
DISPLAY TYPE 'E'.

```

```

        ENDIF.
    ENDLOOP.
*
    REFRESH: gt_cotiza_outtab.
*
    gt_cotiza_outtab[] = gt_cotiza_outtab_aux[].
*
    PERFORM procesa_data.
    ENDIF.
ENDIF.

```

```

    ENDCASE.
ENDIF.
ENDIF.

```

```

WHEN 'BORRAR_NIVEL'.

```

```

    CLEAR: answer.
    CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
        titlebar                = 'Confirmación'
*
        DIAGNOSE_OBJECT         = ' '
        text_question           = '¿Desea borrar el nivel y todos sus c

```

```

ompentas?'
    text_button_1          = 'Sí'
*   ICON_BUTTON_1         = ' '
    text_button_2          = 'No'
*   ICON_BUTTON_2         = ' '
*   DEFAULT_BUTTON        = '1'
    display_cancel_button = 'X'
*   USERDEFINED_F1_HELP   = ' '
*   START_COLUMN          = 25
*   START_ROW             = 6
*   POPUP_TYPE            =
*   IV_QUICKINFO_BUTTON_1 = ' '
*   IV_QUICKINFO_BUTTON_2 = ' '
IMPORTING
    answer                  = answer
*   TABLES
*   PARAMETER              =
EXCEPTIONS
    text_not_found         = 1
    OTHERS                  = 2.
IF sy-subrc <> 0.
* Implement suitable error handling here
ENDIF.

CHECK answer EQ '1'.

CALL METHOD grid_cotiza->get_selected_cells
IMPORTING
    et_cell = lt_cells.

READ TABLE lt_cells INTO w_cell INDEX 1.

IF sy-subrc = 0.
    READ TABLE gt_cotiza_outtab_alv INTO wa_cotiza_outtab_alv INDEX
w_cell-row_id.
    IF sy-subrc = 0 AND wa_cotiza_outtab_alv-
st_fila <> '.'"wa_cotiza_outtab_alv-st_fila NE '2'.
        CASE w_cell-col_id.
            WHEN 'TITULO'.
                IF wa_cotiza_outtab_alv-titulo IS NOT INITIAL.

```

```

        READ TABLE gt_cotiza_outtab INTO wa_cotiza_outtab INDEX
w_cell-row_id.
        IF wa_cotiza_outtab-st_fila = '1'.
            DELETE gt_cotiza_outtab WHERE titulo = wa_cotiza_outt
ab-titulo.
        ELSE.
            LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza_outtab>
WHERE titulo = wa_cotiza_outtab-titulo.
                <fs_cotiza_outtab>-st_fila = '2'.
                <fs_cotiza_outtab>-icon = '@11@'.
            ENDLOOP.
        ENDIF.
    ENDIF.
    WHEN 'SUBTITULO'.
        IF wa_cotiza_outtab_alv-subtitulo IS NOT INITIAL.

            ENDIF.

        WHEN 'CARACTERICA'.
            IF wa_cotiza_outtab_alv-caracterica IS NOT INITIAL.

                DELETE gt_cotiza_outtab WHERE titulo = wa_cotiza_outt
ab-titulo AND
                                                    subtitulo = wa_cotiza_o
uttab-subtitulo AND
                                                    caracterica = wa_cotiza
_outtab-caracterica.
            ELSE.
                LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza_outtab>
WHERE titulo = wa_cotiza_outtab-titulo AND
subtitulo = wa_cotiza_outtab-subtitulo AND
caracterica = wa_cotiza_outtab-caracterica.
                    <fs_cotiza_outtab>-st_fila = '2'.
                    <fs_cotiza_outtab>-icon = '@11@'.
                ENDLOOP.
            ENDIF.

```

```

ENDIF.

WHEN 'SUBCARACTERISTICA'.
    IF wa_cotiza_outtab_alv-subcaracteristica IS NOT INITIAL.

        IF wda_cotiza_outtab-st_fila = '1'.
            DELETE gt_cotiza_outtab WHERE titulo = wa_cotiza_outt
ab-titulo AND
                                                    subtitulo = wa_cotiza_o
uttab-subtitulo AND
                                                    caracterica = wa_cotiza
_outtab-caracterica AND
                                                    subcaracteristica = wa_
cotiza_outtab-subcaracteristica.
        ELSE.
            LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza_outtab>
WHERE titulo = wa_cotiza_outtab-titulo AND

subtitulo = wa_cotiza_outtab-subtitulo AND

caracterica = wa_cotiza_outtab-caracterica AND

subcaracteristica = wa_cotiza_outtab-subcaracteristica.
            <fs_cotiza_outtab>-st_fila = '2'.
            <fs_cotiza_outtab>-icon = '@11@'.
            ENDLOOP.
        ENDIF.

    ENDIF.

ENDCASE.
PERFORM procesa_data.
ENDIF.
ENDIF.

```

```

        ENDCASE.

*      ENDIF.
<fs_cotiza> TYPE LINE OF gt_cotiza_outtab,
            <fs_cal>    TYPE any,
            <fs_val>    TYPE any,
            <config>    TYPE any..
DATA: nombre          TYPE string,
      nombre_config TYPE string.

DATA:lv_contador(3) TYPE n VALUE '000',
      l_compara..

LOOP AT  edr_data1_changed->mlet_mod_cells ASSIGNING <fs_cel> .

      PERFORM valor_nuevo TABLES t_plantillas1_aux  t_plantillas2_aux USI
NG      <fs_cel>-fieldname <fs_cel>-value
            <fs_cel> .

*      READ TABLE gt_cotiza_outtab ASSIGNING <fs_cotiza> INDEX <fs_cel>-
row_id.
*      IF sy-subrc EQ 0.
*
*      CASE <fs_cel>-fieldname.
*      WHEN 'VALOR'.

```



```

*
*
*
*
*          READ TABLE t_config INTO wa_config WITH KEY titulo = <fs_c
otiza>-titulo
*
*                                     subtit = <fs_cotiza>-
subtitulo
*
*                                     caract = <fs_cotiza>-
caracterica
*
*                                     subcar = <fs_cotiza>-
subcaracteristica.
*
*          IF sy-subrc = 0.
*
*          IF wa_config-columna < 80.
*
*          CONCATENATE 'wa_plantilla1-col' wa_config-
columna INTO nombre.
*
*          CONCATENATE 'WA_FLAG1-COL' wa_config-
columna INTO nombre_config.
*
*          ASSIGN (nombre_config) TO <config>.
*
*          ASSIGN (nombre) TO <fs_cal>.
*
*
*          ELSE.
*
*          CONCATENATE 'wa_plantilla2-col' wa_config-
columna INTO nombre.
*
*          CONCATENATE 'WA_FLAG2-COL' wa_config-
columna INTO nombre_config.
*
*          ASSIGN (nombre_config) TO <config>.
*
*          ASSIGN (nombre) TO <fs_cal>.
*
*
*          ENDIF.
*
**
**          -----
*
*
**          -----
*
*
*
*
*          *          <config> = ''.
*
*          ELSE.

```

```

*
*
*      READ TABLE t_plantillas1 INTO wa_plantilla1 WITH KEY
*      dmp200_m_text = wa_plantilla2-texto.
*      DATA: cont(3) TYPE n,
*             colum(3) TYPE n.
*      cont = wa_config-columna - 1.
*      colum = 100 - cont.
*      DO colum TIMES.
*          ADD 1 TO cont.
*          IF cont < 80.
*              CONCATENATE 'wa_flag1-col' cont INTO nombre.
*              ASSIGN (nombre) TO <fs_val>.
*              <fs_val> = 'X'.
*
*          ELSE.
*              CONCATENATE 'wa_flag2-col' cont INTO nombre.
*              ASSIGN (nombre) TO <fs_val>.
*              <fs_val> = 'X'.
*
*          ENDIF.
*      ENDDO.
*
*      ENDIF.
*
*      ENDIF.
*
*      <fs_cotiza>-valor = <fs_cel>-value.
*      <fs_cal> = <fs_cel>-value.
**
*      ENDCASE.
*      ENDIF.
*
*      ENDIF.
ENDLOOP.
*
*      PERFORM formatear_plantilla TABLES gt_cotiza_outtab.
*      PERFORM procesa_data.
*      grid_cotiza-> *          OTHERS          = 2
*      ).
*
*
*

```

```

*
*   grid_cotiza->get_scroll_info_via_id(
*   importing
*       es_row_no   = es_row_no
*       es_row_info = es_row_info
*       es_col_info = es_col_info
*   ).

```

ENDMETHOD.

ENDCLASS.

Declaración de funciones: ZPPP_COTIZACION_FINAL_F01-----

```

*&-----*
*&   Include           ZPPP_COTIZACION_F01
*&-----*
*&-----*
*&   Form   INICIALIZAR_VALORES
*&-----*
*       text
*-----*
*   --> p1           text
*   <-- p2           text
*-----*

```

FORM inicializar_valores .

```

SELECT * FROM ztpc_claves INTO TABLE gt_claves.
SELECT kunnr name1 name2 FROM kna1 INTO TABLE t_clientes.
SELECT pernr nchmc vnamc FROM m_premn INTO CORRESPONDING FIELDS OF TABLE t_vendedores.

```

```

SELECT * FROM tinct INTO TABLE t_incoterms WHERE spras = 'S'.

```

```

SELECT * FROM t052u INTO TABLE t_t052u WHERE spras = 'S'.

```

```

SELECT * FROM ztpc_constant INTO TABLE t_tip_placa

```

```

WHERE modulo = 'PP' AND
      aplicacion = 'COTIZACION' AND
      programa   = 'TIPO_PLACA' AND
      campo      = 'LICENSE_PLATE'.

```

```

DELETE ADJACENT DUPLICATES FROM t_tip_placa COMPARING valor2.

```

```

*t_tip_placa

```

ENDFORM.

```

*&-----*

```

```

*&      Form  CONSULTAR_PLANTILLAS
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM consultar_plantillas .

```

```

gv_cod_ref2      = wa_output-codigo.
gv_secuen_ref2   = wa_output-secuen_config.
dmp200_tip_vta   = wa_output-tip_vta.
dmp200_matnr     = wa_output-matnr.
dmp200_mod_bus   = wa_output-mod_bus.
dmp200_tip_car   = wa_output-tip_car.
dmp200_vers      = wa_output-vers.
dmp200_ano_mod   = wa_output-ano.
PERFORM obtener_plantilla USING dmp200_producto.

```

ENDFORM.

```

*&-----*
*&      Form  LOAD_PIC_FROM_DB
*&-----*
*      text
*-----*
*      <--P_URL text
*-----*
FORM load_pic_from_db CHANGING url.

```

```

*
DATA query_table TYPE w3query OCCURS 1 WITH HEADER LINE.
DATA html_table  TYPE w3html  OCCURS 1.
DATA return_code TYPE w3param-ret_code.
DATA content_type TYPE w3param-cont_type.
DATA content_length TYPE w3param-cont_len.
DATA pic_data      TYPE w3mime OCCURS 0.
DATA pic_size      TYPE i.

REFRESH query_table.
query_table-name = '_OBJECT_ID'.
* query_table-value = 'Z_PRUEBA_DYNPRO'." 'Z_SEUZ_1'. "Nuestra imagen
query_table-value = gv_imagen.

```

```

* query_table-value = 'Z_BUSS4'.

*
IF gv_imagen = 'Z_BUS_NO_DISP'.
    CLEAR gv_imagen.
ENDIF.
ENDFORM.

FORM cargar_fieldcat_detalle .

FIELD-
SYMBOLS: <fs_fieldcatalog_cotiza> TYPE LINE OF gt_fieldcatalog_cotiza.

IF sy-subrc <> 0.
* Implement suitable error handling here
ENDIF.
LOOP AT gt_fieldcatalog_cotiza ASSIGNING <fs_fieldcatalog_cotiza>.
    <fs_fieldcatalog_cotiza>-outputlen = 20.
    CASE <fs_fieldcatalog_cotiza>-fieldname.
        WHEN 'ICON'.
            <fs_fieldcatalog_cotiza>-outputlen = 4.
        WHEN 'ST_FILA'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
        WHEN 'COD_HP'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.

        WHEN 'POSICION'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
        WHEN 'CODIGO'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
        WHEN 'SECUEN_CONFIG'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
        WHEN 'ORD_TIT'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
        WHEN 'ORD_SUBT'.
            <fs_fieldcatalog_cotiza>-no_out = 'X'.
*
        ENDIF.
        WHEN 'ORD_CARAC'.

```

```

    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'ORD_SUBCA'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'USUARIO'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'FECH_CREA'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'HOR_CREA'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'COD_REF2'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'SECUEN_REF2'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'FLG_PREC_CAR'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'NUM_COT'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'PREC_CARAC'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'PREC_TIT'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'MONEDA'.
    <fs_fieldcatalog_cotiza>-no_out = 'X'.
WHEN 'VALOR'.
    <fs_fieldcatalog_cotiza>-f4availabl = 'X'.
*   WHEN 'MATCH'.
*       <fs_fieldcatalog_cotiza>-no_out = 'X'.
*   ENDIF.
WHEN OTHERS.
ENDCASE.
ENDLOOP.
ENDFORM.

FORM inicializar_layout_detalle.
    gs_layout_cotiza-col_opt = 'X'.
*   gs_layout_cotiza-sel_mode = 'D'.
    gs_layout_cotiza-stylefname = 'FIELD_STYLE'.
    gs_layout_cotiza-ctab_fname = 'FIELD_COL'.

ENDFORM.

```

FORM exclude_buttons_detalle .

ENDFORM.

FORM catalogo .

REFRESH: it_catalogo.

* PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'SECUEN_CONFIG' 'Config.' 'X' 8 ' ' ' '.

* PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'CODIGO' 'Cod. Config' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'TIP_VTA' 'Tip.Vta.' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'ANO' 'Año' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MOD_BUS' 'Modelo Bus' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'TIP_CAR' 'Tipo Carroceria' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'VERS' 'Versión' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MAR_CHA' 'Marca Chasis' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MOD_CHA' 'Modelo Chasis' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'COBUST' 'Combustible' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MOTOR' 'Motor' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MOD_MOTOR' 'Modelo Motor' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'MATNR' 'Material' 'X' 14 ' ' ' '.

* PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'PV_NETO' 'Costo. Mat.' 'X' 14 ' ' ' '.

* PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'COST_SERV' 'Cost. Serv' 'X' 14 ' ' ' '.

* PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'COST_MAN_OB' 'Cost. Mano Ob.' 'X' 14 ' ' ' '.

PERFORM agregar_catalogo USING 'GT_PLANTILLAS_ST' 'STANDARD' 'Predete

```
rminada'      'X' 14 ' ' ' '.
```

```
ENDFORM.
```

```
FORM agregar_catalogo USING tabname fieldname coltext do_sum outputle  
n ref_table ref_field .
```

```
wa_catalogo-tabname      = tabname      .  
wa_catalogo-fieldname    = fieldname    .  
wa_catalogo-coltext      = coltext      .  
wa_catalogo-do_sum       = do_sum       .  
wa_catalogo-outputlen    = outputlen    .  
wa_catalogo-ref_field    = ref_field    .  
wa_catalogo-ref_table    = ref_table    .
```

```
* IF fieldname = 'ORDEN'.  
*     wa_catalogo2-no_out = 'X'.  
* ELSE.  
*     ENDIF.
```

```
APPEND wa_catalogo TO it_catalogo.
```

```
CLEAR wa_catalogo.
```

```
ENDFORM.
```

```
FORM valor_editable USING      p_wa_config_titulo  
                                p_wa_config_subtitulo  
                                p_wa_config_caracterica  
                                p_wa_config_subcaracteristica  
                                CHANGING p_flag_editable.
```

```
DATA: cant_rep TYPE i.
```

```
CLEAR cant_rep.
```

```
*gt_cotiza_outtab
```

```
CLEAR p_flag_editable.
```

```
cant_rep = 0.
```

```
IF p_wa_config_titulo  
    p_wa_config_subtitulo      IS INITIAL AND  
    p_wa_config_caracterica    IS INITIAL AND  
    p_wa_config_subcaracteristica IS INITIAL.
```

```
LOOP AT gt_cotiza_outtab WHERE titulo = p_wa_config_titulo.
```

```
    cant_rep = cant_rep + 1.
```



```

    IF cant_rep >= 2.
*   p_flag_editable = 'X'.
        EXIT.
    ENDIF.
ENDLOOP.
cant_rep = 5.

ELSEIF p_wa_config_titulo
    p_wa_config_subtitulo
    p_wa_config_caracterica          IS INITIAL AND
    p_wa_config_subcaracteristica IS INITIAL.

LOOP AT gt_cotiza_outtab WHERE titulo = p_wa_config_titulo AND
                                subtitulo = p_wa_config_subtitulo.

    cant_rep = cant_rep + 1.
    IF cant_rep >= 2.
        EXIT.
    ENDIF.
ENDLOOP.

ELSEIF p_wa_config_titulo
    p_wa_config_subtitulo
    p_wa_config_caracterica
    p_wa_config_subcaracteristica .

LOOP AT gt_cotiza_outtab WHERE titulo = p_wa_config_titulo AND
                                subtitulo = p_wa_config_subtitulo AND
                                caracterica = p_wa_config_caracterica.

    cant_rep = cant_rep + 1.
    IF cant_rep >= 2.
        EXIT.
    ENDIF.
ENDLOOP.

ELSEIF p_wa_config_titulo
    p_wa_config_subtitulo
    p_wa_config_caracterica
    p_wa_config_subcaracteristica .

LOOP AT gt_cotiza_outtab WHERE titulo = p_wa_config_titulo AND
                                subtitulo = p_wa_config_subtitulo AND
                                caracterica = p_wa_config_caracterica

```

AND

subcaracteristica = p_wa_config_subcar

acteristica.

cant_rep = cant_rep + 1.

IF cant_rep >= 2.

EXIT.

ENDIF.

ENDLOOP.

ENDIF.

IF cant_rep < 2.

p_flag_editable = 'X'.

ELSE.

CLEAR cant_rep.

ENDIF.

ENDFORM.

```
*&-----*
*&      Form  TIPO_LINEA
*&-----*
*      text
*-----*
*      -->P_WA_CONFIG  text
*      <--P_TIP_LINEA  text
*-----*
```

FORM tipo_linea USING p_wa_config TYPE gty_cotiza_outtab
CHANGING p_tip_linea.

IF p_wa_config-titulo

p_wa_config-subtitulo

p_wa_config-caracterica

p_wa_config-subcaracteristica .

p_tip_linea = 1.

ELSEIF p_wa_config-titulo

p_wa_config-subtitulo

p_wa_config-caracterica

p_wa_config-subcaracteristica .

```

p_tip_linea = 2.

ELSEIF p_wa_config-titulo
    p_wa_config-subtitulo
    p_wa_config-caracterica
    p_wa_config-subcaracteristica .

p_tip_linea = 3.

ELSEIF p_wa_config-titulo
    p_wa_config-subtitulo
    p_wa_config-caracterica
    p_wa_config-subcaracteristica .
p_tip_linea = 4.

ENDIF.

ENDFORM.

*&-----*
*&      Form  RENUMERAR_TABLA
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*

FORM renumerar_tabla USING posicion suma .
    FIELD-SYMBOLS: <fs_cotiza_alv> TYPE LINE OF wa_cotiza_outtab_alv,
                  <fs_cotiza>     TYPE LINE OF wa_cotiza_outtab_alv.

    DATA: cant TYPE i.
    CLEAR  cant.
    LOOP AT gt_cotiza_outtab_alv ASSIGNING <fs_cotiza_alv> .
        IF cant < <fs_cotiza_alv>-ord_tit.
            cant = <fs_cotiza_alv>-ord_tit.
            IF <fs_cotiza_alv>-ord_tit > posicion.
                ADD suma TO <fs_cotiza_alv>-ord_tit.
            ENDIF.
        ENDIF.
    ENDLOOP.

```

```

ELSE.
    EXIT.
ENDIF.
ENDLOOP.

CLEAR cant.
LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza> .
    IF cant < <fs_cotiza>-ord_tit.
        cant = <fs_cotiza>-ord_tit.
        IF <fs_cotiza>-ord_tit > posicion.
            ADD suma TO <fs_cotiza>-ord_tit.
        ENDIF.
    ELSE.
        EXIT.
    ENDIF.
ENDLOOP.

SORT gt_cotiza_outtab ASCENDING BY ord_tit.
SORT gt_cotiza_outtab_alv ASCENDING BY ord_tit.

ENDFORM.

FORM procesa_data .
    FIELD-SYMBOLS: <fs_cotiza> TYPE LINE OF gt_cotiza_outtab.
    DATA: wa_cotiza TYPE LINE OF gt_cotiza_outtab.
    DATA: dmpv_tiplinea TYPE i.
    DATA: dmpv_flete TYPE ze_pp_pv_final.
    DATA: dmpv_dcto TYPE ze_pp_pv_final.
    DATA: dmpv_vkorg TYPE vkorg.

    CLEAR: lv_vkorg.

*PERFORM limpiar_montos.

*READ TABLE gt_cotiza_outtab WITH KEY ST_FILA = '1'.

READ TABLE get_cotiza_outtab ASSIGNING <fs_cotiza> WITH KEY titulo
= 'CHASIS'

```

subtitulo

```

= ''
caracteric

a = ''
subcaracte

ristica = ''
edita = ''

.

IF sy-subrc <> 0.

name = '%#AUTOTEXT036'.

ELSE.

*   IF dmp200_cost_chasis IS INITIAL.
*       dmp200_chasis = <fs_cotiza>-prec_tit.
dmp200_cost_chasis2 = <fs_cotiza>-prec_tit.
*   ELSE.
*       <fs_cotiza>-prec_tit = dmp200_cost_chasis2.
*   ENDIF.

SORT gt_cotiza_outtab ASCENDING BY ord_tit.
LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza>.

    PERFORM valor_editable USING <fs_cotiza>-titulo
                                <fs_cotiza>-subtitulo
                                <fs_cotiza>-caracterica
                                <fs_cotiza>-subcaracteristica
                                CHANGING <fs_cotiza>-edita.

    IF ge_ucomm = 'V'.
        CLEAR <fs_cotiza>-edita.
    ENDIF.

ENDLOOP.

```

```

CLEAR dmp200_precio.
REFRESH gt_cotiza_outtab_alv.
LOOP AT gt_cotiza_outtab ASSIGNING <fs_cotiza>.
  REFRESH <fs_cotiza>-field_style.
*   IF <fs_cotiza>-edita = 'X' AND <fs_cotiza>-st_fila = 1.

*   ENDIF.

  IF <fs_cotiza>-edita = 'X' AND ( ge_ucomm = 'M' OR ge_ucomm
= 'N' OR ge_ucomm = 'F' ) AND <fs_cotiza>-subtitulo <> 'MARCA-MODELO-
CHASIS' AND
<fs_cotiza>-subtitulo <> 'MARCA-MODELO-MOTOR' AND
<fs_cotiza>-subtitulo <> 'COMBUSTIBLE' AND
<fs_cotiza>-subtitulo <> 'FORMULA RODANTE'.
    ELSE.
      ls_stylerow-fieldname = 'VALOR' .
      CLEAR: ls_stylerow.
    ENDIF.

PERFORM tipo_linea USING <fs_cotiza> CHANGING lv_tiplinea.
PERFORM agrega_color USING lv_tiplinea CHANGING <fs_cotiza> .

CLEAR wa_cotiza.
MOVE <fs_cotiza> TO wa_cotiza.

IF lv_tiplinea = 1 AND <fs_cotiza>-titulo NE 'CHASIS'.
  ADD <fs_cotiza>-prec_tit TO dmp200_precio.
ENDIF.

IF lv_tiplinea = 2.
  CLEAR: wa_cotiza-titulo.
ENDIF.

IF lv_tiplinea = 3.
  CLEAR: wa_cotiza-titulo,
        wa_cotiza-subtitulo.
ENDIF.

IF lv_tiplinea = 4.
  CLEAR: wa_cotiza-titulo,
        wa_cotiza-subtitulo,

```

```

        wa_cotiza-caracterica.
    ENDIF.

    APPEND wa_cotiza TO gt_cotiza_outtab_alv.

ENDLOOP.

* READ TABLE gt_cotiza_outtab WITH KEY ST_FILA = '1'.
*if sy-subrc = 0.
* PERFORM limpiar_montos.
* exit.
* ENDIF.
CLEAR: dmp200_pv_final,dmp200_pv_igv.
CLEAR dmp200_dcto.
CLEAR lv_flete.

CLEAR lv_xcpdk.
* dmp200_costo_total
* dmp200_tip_placa = gv_tip_placa.
IF p_clie IS NOT INITIAL.
    dmp200_kunnr = p_clie.
    IF dmp200_kunnr2 IS INITIAL.
        dmp200_kunnr2 = dmp200_kunnr.
    ENDIF.
* dmp200_kunnr2 = p_clie.
ENDIF.
* dmp200_vendedor = p_vend.
* dmp200_ano_mod = p_ano.
* dmp200_ano_fab = p_ano.
lv_kunnr = dmp200_kunnr.
lv_kunnr2 = dmp200_kunnr2.
lv_pernr = dmp200_vendedor.

CLEAR: dmp200_l_vendedor.

IF lv_pernr IS NOT INITIAL.
    SELECT SINGLE nchmc vnamc FROM m_premn INTO (lv_nchmc,lv_vnamc) WHERE
pernr = lv_pernr.
    CONCATENATE lv_nchmc lv_vnamc INTO dmp200_l_vendedor SEPARATED BY spa

```

```

ce.
  ENDIF.
  IF lv_kunnr IS NOT INITIAL.
*   SELECT pernr nchmc vnamc FROM M_PREMN INTO CORRESPONDING FIELDS OF
TABLE t_vendedores.
    SELECT SINGLE land1 name1 name2 FROM knal INTO (dmp200_land1,lv_name1
,lv_name2) WHERE kunnr = lv_kunnr.
    CONCATENATE lv_name1 lv_name2 INTO dmp200_l_kunnr SEPARATED BY space.
*   IF dmp200_land1 = 'PE'.
*     lv_vkorg = 'MD01'.
*   ELSE.
*     lv_vkorg = 'MD02'.
*   ENDIF.
*   IF dmp200_zterm

*     vkorg = lv_vkorg AND
*     vtweg = 'MB' AND
*     spart = '99'.
*   ENDIF.

ENDIF.
SELECT SINGLE text1 FROM t052u INTO dmp200_l_zterm WHERE
  spras = 'S' AND
  zterm = dmp200_zterm.
CLEAR dmp200_l_incoterm.
READ TABLE t_incoterms WITH KEY incol = dmp200_incoterm.
IF sy-subrc = 0.
  dmp200_l_incoterm = t_incoterms-bezei.
ENDIF.

IF lv_kunnr2 IS NOT INITIAL.
*   SELECT pernr nchmc vnamc FROM M_PREMN INTO CORRESPONDING FIELDS OF
TABLE t_vendedores.
    SELECT SINGLE land1 name1 name2 xcpdk FROM knal INTO (dmp200_land1,lv
_name1,lv_name2,lv_xcpdk) WHERE kunnr = lv_kunnr2.
    IF lv_xcpdk = 'X' AND lv_kunnr2 = lv_kunnr.
      dmp200_l_kunnr2 = dmp300_c_nombre.
    ELSE.
      CONCATENATE lv_name1 lv_name2 INTO dmp200_l_kunnr2 SEPARATED BY spa
ce.
    ENDIF.
  ENDIF.

```



```

IF dmp200_land1 = 'PE'.
    lv_vkorg = 'MD01'.
ELSE.
    lv_vkorg = 'MD02'.
ENDIF.

IF dmp200_zterm IS INITIAL.
    SELECT zterm FROM knvv INTO dmp200_zterm WHERE
        kunnr = lv_kunnr2 AND
        vkorg = lv_vkorg AND
        vtweg = 'MB' AND
        spart = '99'.
ENDIF.

IF dmp200_land1 IS NOT INITIAL.
    ENDIF.
*   dmp200_l_kunnr2 = dmp200_l_kunnr.
ENDIF.
*

IF gt_cotiza_outtab[] IS NOT INITIAL.
*   dmp200_moneda = 'PEN'.
    dmp200_cost_carroc = dmp200_precio + gv_cost_man_ob + gv_cost_serv +
gv_cost_seguridad.
    dmp200_cost_chasis = dmp200_cost_chasis2.
*   dmp200_cost_chasis = dmp200_cost_chasis * dmp200_cant.
*   dmp200_cost_carroc = dmp200_cost_carroc * dmp200_cant.

    IF c_pen = 'X' AND p_t_camb IS NOT INITIAL.
        dmp200_cost_chasis = dmp200_cost_chasis * p_t_camb.
        dmp200_cost_carroc = dmp200_cost_carroc * p_t_camb.
    ENDIF.

    dmp200_cost_chasis2 = dmp200_cost_chasis.

    dmp200_prec_carroc = dmp200_cost_carroc / ( 1 - p_marg / 100 ).
    dmp200_prec_chasis = dmp200_cost_chasis / ( 1 - dmp200_margen / 100
).
    dmp200_flete = p_flete.

    dmp200_prec_netto = dmp200_prec_carroc + dmp200_prec_chasis + p_flet
e.

```

```

IF dmp200_land1 = 'PE' OR dmp200_kunnr2 IS INITIAL.
    "02/03/2017
*      dmp200_igv = ( dmp200_prec_netto + dmp200_flete ) * 18 / 100. "com
ent DB
    dmp200_igv = ( dmp200_prec_netto ) * 18 / 100. "coment DB
ELSE.
    CLEAR dmp200_igv.
ENDIF.

dmp200_prec_bruto = dmp200_prec_netto + dmp200_igv.

PERFORM calcular_participacion_margen
USING dmp200_cost_chasis dmp200_cost_carroc dmp200_prec_chasis dmp200
_prec_carroc
CHANGING dmp200_particip_chasis dmp200_particip_carroc .

* IF r_por = 'X'.
*      dmp200_dcto_f = dmp200_pv_final * p_descu / 100.
* ELSE.
    dmp200_dcto_f = p_descu.
* ENDIF.
DATA: dmpv_chasis TYPE ze_pp_dcto.

dmp200_flete = p_flete.
dmp200_flete_f = p_flete.

dmp200_prec_bruto_f = dmp200_prec_bruto / 100 .

dmp200_prec_bruto_f = ceil( dmp200_prec_bruto_f ).

```

```

dmp200_prec_bruto_f = dmp200_prec_bruto_f * 100.

dmp200_prec_venta_f = dmp200_prec_bruto_f - dmp200_dcto_f .
IF dmp200_land1 = 'PE' OR dmp200_kunnr2 IS INITIAL.
    dmp200_igv_f = dmp200_prec_venta_f * 18 / 118.
ELSE.
    CLEAR dmp200_igv_f.
ENDIF.

PERFORM calcular_margen USING dmp200_particip_chasis dmp200_particip_
carroc dmp200_cost_carroc dmp200_cost_chasis CHANGING dmp200_particip_cha
sis_f dmp200_particip_carroc_f .

dmp200_pv_total = dmp200_prec_venta_f * dmp200_cant.

dmp200_prec_netto_f = dmp200_prec_venta_f - dmp200_igv_f - dmp200_flet
e_f.

ENDIF.
* PERFORM calcular_margen USING dmp200_particip_chasis dmp200_particip_c
arrocc dmp200_cost_carroc dmp200_cost_chasis CHANGING dmp200_particip_chas
is_f dmp200_particip_carroc_f .
*ENDIF.

LOOP AT SCREEN.
    IF ge_ucomm = 'C' OR ge_ucomm = 'M'.
        IF screen-name = 'DMP200_L_KUNNR' OR screen-
name = 'DMP200_L_KUNNR2' OR screen-name = 'DMP200_KUNNR'.
            IF dmp200_kunnr IS INITIAL.
                screen-input = '1'
            ENDIF.
        ENDLOOP.
    ENDFORM.
GRABAR_COTIZACION

FORM grabar_cotizacion .
    DATA: dmpv_flag_solicitud.
    DATA: dmpv_mensaje(180) TYPE c.
    DATA: lwa_cab_cotizacion TYPE ztpp_cotiza_cab.
    DATA: lt_cab_longitud TYPE ztpp_longitud.
    DATA: lt_det_cotizacion TYPE STANDARD TABLE OF zpp_cotiza_det WITH HEAD

```

```

ER LINE.
FIELD-SYMBOLS: <lfs_cotiza> TYPE LINE OF gt_cotiza_outtab.
DATA: l_number(10) TYPE c.
DATA: dmpv_posicion TYPE zpp_cotiza_det-posicion.
FIELD-SYMBOLS: <fs> TYPE any.
DATA: i(3) TYPE n.
DATA: dmpv_nom_campo TYPE string.
DATA: lwa_cotiza_det1 TYPE ztpp_cotiza_det1.
DATA: lwa_cotiza_det2 TYPE ztpp_cotiza_det2.
DATA: lwa_cotiza_st1 TYPE ztpp_cotiza_st1.
DATA: lwa_cotiza_st2 TYPE ztpp_cotiza_st2.
DATA: lwa_cotiza_conf TYPE STANDARD TABLE OF ztpp_cotiza_conf WITH HEAD
ER LINE.
CLEAR: lwa_cotiza_det1,lwa_cotiza_det2,lwa_cotiza_st1,lwa_cotiza_st2,lw
a_cotiza_conf.
CLEAR lv_posicion.

i = '001'.
DO 100 TIMES.
  IF i < 80 .
    lv_nom_campo = 'wa_flag1-col'.
  ELSE.
    lv_nom_campo = 'wa_flag2-col'.
  ENDIF.
  CONCATENATE lv_nom_campo i INTO lv_nom_campo.
  CONDENSE lv_nom_campo.
  ASSIGN (lv_nom_campo) TO <fs>.
  IF <fs> = ''.
    lv_flag_solicitud = 'X'.
    EXIT.
  ENDIF.
  ADD 1 TO i.
ENDDO.

IF dmp200_numcot IS INITIAL.
*   LOOP AT gt_cotiza_outtab WHERE st_fila NE ''.
*     lv_flag_solicitud = 'X'.
*     EXIT.
*   ENDLOOP.

```

"CABECERA DE COTIZACION

```

PERFORM genera_num_cotizacion USING l_number.
IF l_number IS NOT INITIAL.
    dmp200_numcot                = l_number.
    lwa_cab_cotizacion-num_cot   = dmp200_numcot.

    lwa_cab_cotizacion-cod_ref   = dmp200_producto.
    lwa_cab_cotizacion-kunnr     = dmp200_kunnr.
    lwa_cab_cotizacion-kunnr2    = dmp200_kunnr2.
    lwa_cab_cotizacion-landl     = dmp200_landl.

    lwa_cab_cotizacion-zterm     = dmp200_zterm.
    lwa_cab_cotizacion-vendedor  = dmp200_vendedor.
    lwa_cab_cotizacion-tip_vta   = dmp200_tip_vta.
    lwa_cab_cotizacion-matnr     = dmp200_matnr.
    lwa_cab_cotizacion-mod_bus   = dmp200_mod_bus.
    lwa_cab_cotizacion-tip_car   = dmp200_tip_car.
    lwa_cab_cotizacion-cantidad  = dmp200_cant.
    lwa_cab_cotizacion-ano_mod   = dmp200_ano_mod.
    lwa_cab_cotizacion-ano_fab   = dmp200_ano_fab.
    lwa_cab_cotizacion-ano_fab_carr = dmp200_ano_fab_carr.
    lwa_cab_cotizacion-usuario   = sy-uname.
    IF c_usd = 'X'.
        lwa_cab_cotizacion-moneda = 'USD'.
    ELSEIF c_pen = 'X' .
        lwa_cab_cotizacion-moneda = 'PEN'.
    ENDIF.

    lwa_cab_cotizacion-t_camb     = p_t_camb.
    lwa_cab_cotizacion-tip_placa  = dmp200_tip_placa.
    lwa_cab_cotizacion-observ_cotiza = dmp200_observ_cotiza.
    lwa_cab_cotizacion-longitud  = dmp200_long.
    lwa_cab_cotizacion-alto      = dmp200_alto.
    lwa_cab_cotizacion-vol_del   = dmp200_vol_del.
    lwa_cab_cotizacion-ancho     = dmp200_ancho.
    lwa_cab_cotizacion-dee       = dmp200_dee.
    lwa_cab_cotizacion-vol_post  = dmp200_vol_post.
    lwa_cab_cotizacion-pv_netto  = dmp200_prec_netto_f.
    lwa_cab_cotizacion-flete     = dmp200_flete_f.
    lwa_cab_cotizacion-igv       = dmp200_igv_f.
    lwa_cab_cotizacion-cost_man_ob = gv_cost_man_ob.
    lwa_cab_cotizacion-cost_serv  = gv_cost_serv.
    lwa_cab_cotizacion-dcto      = dmp200_dcto_f.
    lwa_cab_cotizacion-pv_final  = dmp200_pv_total.

```

```

lwa_cab_cotizacion-marg_ganancha_r = dmp200_margen.
lwa_cab_cotizacion-marg_gananc_r   = p_marg.
lwa_cab_cotizacion-marg_ganancha   = dmp200_particip_chasis_f.
lwa_cab_cotizacion-marg_gananc     = dmp200_particip_carroc_f.
lwa_cab_cotizacion-ruc              = dmp300_c_ruc.
lwa_cab_cotizacion-dni              = dmp300_c_dni.
lwa_cab_cotizacion-nombre_cli       = dmp300_c_nombre.
lwa_cab_cotizacion-contac           = dmp300_c_contacto.
lwa_cab_cotizacion-correo           = dmp300_c_correo.
lwa_cab_cotizacion-telefono         = dmp300_c_telefono.
lwa_cab_cotizacion-direccion        = dmp300_c_direccion.
lwa_cab_cotizacion-prec_chasis     = dmp200_prec_chasis.
lwa_cab_cotizacion-prec_carro      = dmp200_prec_carroc.
*  lwa_cab_cotizacion- FLETE         = DMP200_FLETE
lwa_cab_cotizacion-pv_netto_r       = dmp200_prec_netto.
lwa_cab_cotizacion-igv_r            = dmp200_igv.
lwa_cab_cotizacion-incoterm         = dmp200_incoterm .
lwa_cab_cotizacion-kunnr2_l        = dmp200_l_kunnr2.
lwa_cab_cotizacion-vendedor_l      = dmp200_l_vendedor.
lwa_cab_cotizacion-tp_cotiza       = dmp200_tip_cot.
lwa_cab_cotizacion-f_v_cot_d       = dmp200_f_v_cot_d.
lwa_cab_cotizacion-f_v_cot_h       = dmp200_f_v_cot_h.
lwa_cab_cotizacion-plazo_ent       = dmp200_plazo_ent.
lwa_cab_cotizacion-secuen_ref      = dmp200_m_text_s.
lwa_cab_cotizacion-secuen_ref2     = dmp200_m_text.
lwa_cab_cotizacion-confirmado      = dmp200_confirmado.
lwa_cab_cotizacion-cot_ref         = dmp200_numcot_ref.
lwa_cab_cotizacion-fech_crea       = sy-datum.
lwa_cab_cotizacion-hor_crea        = sy-uzeit.
lwa_cab_cotizacion-zalias          = dmp200_moduni.
lwa_cab_cotizacion-cost_seguridad  = gv_cost_seguridad.
"ADD DB7
lwa_cab_cotizacion-version2 = 'X'.

```

"OBTENIENDO DETALLE DE COTIZACION

```

*  MOVE-CORRESPONDING lwa_cotiza_conf TO t_config.
LOOP AT t_config.
  CLEAR lwa_cotiza_conf.
  MOVE-CORRESPONDING t_config TO lwa_cotiza_conf.
  lwa_cotiza_conf-num_cot = dmp200_numcot.

```

```

        APPEND lwa_cotiza_conf.
    ENDLOOP.
    MOVE-CORRESPONDING wa_plantilla1 TO lwa_cotiza_det1.
    MOVE-CORRESPONDING wa_plantilla2 TO lwa_cotiza_det2.
    MOVE-CORRESPONDING wa_flag1 TO lwa_cotiza_st1.
    MOVE-CORRESPONDING wa_flag2 TO lwa_cotiza_st2.

*
    lwa_cotiza_conf-num_cot = dmp200_numcot.
    lwa_cotiza_det1-num_cot = dmp200_numcot.
    lwa_cotiza_det2-num_cot = dmp200_numcot.
    lwa_cotiza_st1-num_cot = dmp200_numcot.
    lwa_cotiza_st2-num_cot = dmp200_numcot.

    MODIFY ztpp_cotiza_cab FROM lwa_cab_cotizacion.
    MODIFY ztpp_cotiza_det1 FROM lwa_cotiza_det1.
    MODIFY ztpp_cotiza_det2 FROM lwa_cotiza_det2.
    MODIFY ztpp_cotiza_st1 FROM lwa_cotiza_st1.
    MODIFY ztpp_cotiza_st2 FROM lwa_cotiza_st2.
    MODIFY ztpp_cotiza_conf FROM TABLE lwa_cotiza_conf[].
    COMMIT WORK AND WAIT.
    PERFORM guarda_estrategias TABLES lt_liberacion USING lv_flag_solici
itud .
    "OBTENIENDO DETALLE DE COTIZACION

*
    LOOP AT gt_cotiza_outtab.
*
        MOVE-CORRESPONDING gt_cotiza_outtab TO lt_det_cotizacion.
*
        ADD 1 TO lv_posicion.
*
        lt_det_cotizacion-posicion = lv_posicion.
*
        lt_det_cotizacion-num_cot = dmp200_numcot.
*
        APPEND lt_det_cotizacion.
*
    ENDLOOP.
*
    MODIFY ztpp_cotiza_cab FROM lwa_cab_cotizacion.
*
    MODIFY zpp_cotiza_det FROM TABLE lt_det_cotizacion[].
    CONCATENATE 'Se Creo la Cotizacion: ' lwa_cab_cotizacion-
num_cot INTO lv_mensaje SEPARATED BY space.
    MESSAGE lv_mensaje TYPE 'S'.
*
    PERFORM envia_correo USING dmp200_numcot.
    ELSE.
        MESSAGE 'Error al Generar Numero de Cotización' TYPE 'S' DISPLAY TY
PE 'E'.
    ENDIF.

```

ELSE.

```
* -----  
  
DATA: dmpv_evaluacion,  
      lv_aceptado.  
* LOOP AT gt_cotiza_outtab WHERE st_fila NE ''.  
*   lv_flag_solicitud = 'X'.  
*   EXIT.  
* ENDLOOP.  
  
* PERFORM guarda_estrategias USING lv_flag_solicitud.  
  
* IF lv_flag_solicitud = 'X'.  
*   lv_evaluacion = '2'.  
* ELSE.  
*   lv_evaluacion = '1'.  
**   lwa_cab_cotizacion-aceptado = 'X'.  
* ENDIF.  
PERFORM save_log.  
  
UPDATE ztpp_cotiza_cab SET  
  
cod_ref          = dmp200_producto  
kunnr           = dmp200_kunnr  
kunnr2          = dmp200_kunnr2  
land1           = dmp200_land1  
zterm           = dmp200_zterm  
vendedor        = dmp200_vendedor  
tip_vta         = dmp200_tip_vta  
matnr           = dmp200_matnr  
mod_bus         = dmp200_mod_bus  
tip_car         = dmp200_tip_car  
cantidad        = dmp200_cant  
ano_mod         = dmp200_ano_mod  
ano_fab         = dmp200_ano_fab  
tip_placa       = dmp200_tip_placa
```


longitud	=	dmp200_long
alto	=	dmp200_alto
vol_del	=	dmp200_vol_del
ancho	=	dmp200_ancho
dee	=	dmp200_dee
vol_post	=	dmp200_vol_post
pv_netto	=	dmp200_prec_netto_f
flete	=	dmp200_flete_f
igv	=	dmp200_igv_f
dcto	=	dmp200_dcto_f
cost_man_ob	=	gv_cost_man_ob
cost_serv	=	gv_cost_serv
pv_final	=	dmp200_pv_total
marg_ganancha_r	=	dmp200_margen
marg_gananc_r	=	p_marg
marg_ganancha	=	dmp200_particip_chasis_f
marg_gananc	=	dmp200_particip_carroc_f
t_camb	=	p_t_camb
ruc	=	dmp300_c_ruc
dni	=	dmp300_c_dni
nombre_cli	=	dmp300_c_nombre
contac	=	dmp300_c_contacto
correo	=	dmp300_c_correo
telefono	=	dmp300_c_telefono
direccion	=	dmp300_c_direccion
prec_chasis	=	dmp200_prec_chasis
prec_carro	=	dmp200_prec_carroc
* FLETE	=	DMP200_FLETE
pv_netto_r	=	dmp200_prec_netto
igv_r	=	dmp200_igv
incoterm	=	dmp200_incoterm
evaluacion	=	lv_evaluacion
* aceptado	=	lv_aceptado
kunnr2_l	=	dmp200_l_kunnr2
vendedor_l	=	dmp200_l_vendedor
tp_cotiza	=	dmp200_tip_cot
f_v_cot_d	=	dmp200_f_v_cot_d
f_v_cot_h	=	dmp200_f_v_cot_h
plazo_ent	=	dmp200_plazo_ent
confirmado	=	dmp200_confirmado
secuen_ref	=	dmp200_m_text_s
secuen_ref2	=	dmp200_m_text

```

*      usuario          = sy-uname
      zalias = dmp200_moduni
      cot_ref = dmp200_numcot_ref
      cost_seguridad = gv_cost_seguridad
      observ_cotiza = dmp200_observ_cotiza
      ano_fab_carr    = dmp200_ano_fab
WHERE  num_cot      = dmp200_numcot.
*      -----

*      DELETE FROM zpp_cotiza_det WHERE num_cot = dmp200_numcot.
*      REFRESH lt_det_cotizacion.
*      LOOP AT gt_cotiza_outtab.
*          MOVE-CORRESPONDING gt_cotiza_outtab TO lt_det_cotizacion.
*          ADD 1 TO lv_posicion.
*          lt_det_cotizacion-posicion = lv_posicion.
*          lt_det_cotizacion-num_cot = dmp200_numcot.
*          APPEND lt_det_cotizacion.
*      ENDLOOP.
*      MODIFY zpp_cotiza_det FROM TABLE lt_det_cotizacion[].

"GUARDAR CAMBIOS
LOOP AT t_config.
    CLEAR lwa_cotiza_conf.
    MOVE-CORRESPONDING t_config TO lwa_cotiza_conf.
    lwa_cotiza_conf-num_cot = dmp200_numcot.
    APPEND lwa_cotiza_conf.
ENDLOOP.
MOVE-CORRESPONDING wa_plantilla1 TO lwa_cotiza_det1.
MOVE-CORRESPONDING wa_plantilla2 TO lwa_cotiza_det2.
MOVE-CORRESPONDING wa_flag1 TO lwa_cotiza_st1.
MOVE-CORRESPONDING wa_flag2 TO lwa_cotiza_st2.

*      lwa_cotiza_conf-num_cot = dmp200_numcot.
      lwa_cotiza_det1-num_cot = dmp200_numcot.
      lwa_cotiza_det2-num_cot = dmp200_numcot.
      lwa_cotiza_st1-num_cot = dmp200_numcot.
      lwa_cotiza_st2-num_cot = dmp200_numcot.

*      MODIFY ztpc_cotiza_cab FROM lwa_cab_cotizacion.
MODIFY ztpc_cotiza_det1 FROM lwa_cotiza_det1.
MODIFY ztpc_cotiza_det2 FROM lwa_cotiza_det2.

```

```

MODIFY ztpp_cotiza_st1 FROM lwa_cotiza_st1.
MODIFY ztpp_cotiza_st2 FROM lwa_cotiza_st2.
MODIFY ztpp_cotiza_conf FROM TABLE lwa_cotiza_conf[].
"GUARDAR CAMBIOS
PERFORM guarda_estrategias TABLES lt_liberacion USING lv_flag_solicitud .
* PERFORM envia_correo USING dmp200_numcot.
CONCATENATE 'Se modificó la Cotizacion: ' dmp200_numcot INTO lv_mensaje SEPARATED BY space.
MESSAGE lv_mensaje TYPE 'S'.

"Grabar Cambios en Cotizac.
ENDIF.
ENDFORM.
*&-----*
*& Form GENERA_NUM_COTIZACION
*&-----*
*
FORM genera_num_cotizacion USING p_l_number.
CALL FUNCTION 'NUMBER_GET_NEXT'
EXPORTING
nr_range_nr = '3'
object = 'ZHP_CODIGO'
* QUANTITY = '1'
* SUBOBJECT = ' '
* TOYEAR = '0000'
* IGNORE_BUFFER = ' '
IMPORTING
number = p_l_number
* QUANTITY =
* RETURNCODE =
EXCEPTIONS

CLEAR p_l_number.
ELSE.

ENDIF.
ENDFORM.
FORM cargar_constantes .
DATA: lwa_constant TYPE ztpp_constant.
SELECT SINGLE * FROM ztpp_constant INTO lwa_constant WHERE modulo = 'P'
AND

```

```

aplicacion = 'COTIZA
CION' AND

programa = 'TIPO_PLA
CA' AND

campo = 'LICENSE_
PLATE' AND

valor1 = p_tcar.

IF sy-subrc = 0.
    gv_tip_placa = lwa_constant-valor2.
ENDIF.
CLEAR lwa_constant.

CLEAR lwa_constant.
SELECT SINGLE * FROM ztpp_constant INTO lwa_constant WHERE modulo = 'P
P' AND

aplicacion = 'COTIZA
CION' AND

programa = 'MARGEN_C
ARROCERI ' AND

campo = 'MRG_GAIN
'.

IF sy-subrc = 0.
    gv_marg_gananc_l = lwa_constant-valor1.
    gv_marg_gananc_h = lwa_constant-valor2.
    p_marg = gv_marg_gananc_l.
ENDIF.

** CLEAR lwa_constant.
** SELECT SINGLE * FROM ztpp_constant INTO lwa_constant WHERE modulo =
'PP' AND

**
aplicacion = 'COTI
ZACION' AND

**
programa = 'MARGEN
_GANANCIA' AND

**
campo = 'MRG_CH
ASIS'.

** IF sy-subrc = 0.
**     gv_marg_ganancha_l = lwa_constant-valor1.
**     gv_marg_ganancha_h = lwa_constant-valor2.
**     p_marg = gv_marg_gananc_h.
**     ENDIF.

```

```

IF p_t_camb IS INITIAL AND c_pen EQ 'X'.
    DATA: ls_tipo_cambio TYPE bapil093_0.
*      "si no ingresan tipo de cambio obtener el tipo de cambio de la fecha
clave
ls_tipo_cambio.

    p_t_camb = ls_tipo_cambio-exch_rate_v.
    v_tpcambio_aux = p_t_camb.
ENDIF.

SELECT SINGLE valor1
    INTO gv_vig_cot
    FROM ztpc_constant
    WHERE modulo = 'PP' AND
           aplicacion = 'COTIZACION' AND
           campo = 'VIGENCIA_COT'.

SELECT SINGLE valor1
    INTO gv_plazo_ent
    FROM ztpc_constant
    WHERE modulo = 'PP' AND
           aplicacion = 'COTIZACION' AND
           campo = 'PLAZO_ENT'.

ENDFORM.

*&-----*
*&      Form  ACTUALIZAR_FLETE
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM actualizar_flete .
    CALL SELECTION-SCREEN 005 STARTING AT 20 5.
ENDFORM.

*&-----*
*&      Form  ACTUALIZAR_TIP_CAMBIO
*&-----*

```

```

*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*

FORM actualizar_tip_cambio .
  CLEAR gv_cancelar_accion.
  CALL SELECTION-SCREEN 006 STARTING AT 20 5.
  IF gv_cancelar_accion = 'X'.
    v_tpcambio_aux = p_t_camb .
  ENDIF.
  IF p_t_camb IS NOT INITIAL.
    DATA: ls_tipo_cambio TYPE bap1093_0.
*   "si no ingresan tipo de cambio obtener el tipo de cambio de la fecha
clave
    IF p_t_camb IS INITIAL OR p_t_camb = 1 AND c_pen = 'X'.
      CALL FUNCTION 'BAPI_EXCHANGERATE_GETDETAIL'
        EXPORTING
          rate_type = 'M'
          from_curr = 'PEN'
          to_currncy = 'USD'
          date       = sy-datum
        IMPORTING
          exch_rate = ls_tipo_cambio.

      p_t_camb = ls_tipo_cambio-exch_rate_v.

    ENDIF.
  ENDIF.

  IF v_tpcambio_aux <> p_t_camb.
    CLEAR: p_flete,p_descu.
  ELSE.
    v_tpcambio_aux = p_t_camb.
  ENDIF.
  IF c_usd = 'X'.
    p_t_camb = '1'.
  ENDIF.
  v_tpcambio_aux = p_t_camb .
* ENDIF.
ENDFORM.

```

```

*&-----*
*&      Form  ACTUALIZAR_MARGEN_GANANC
*&-----*
*      text
*-----*
*  --> p1      text
*  <-- p2      text
*-----*
FORM actualizar_margen_gananc .
    CALL SELECTION-SCREEN 007 STARTING AT 20 5.
    if p_marg is NOT INITIAL.
    *   gv_marg_gananc_h = p_marg.
    *ENDIF.
ENDFORM.

FORM handle_mach USING      p_e_fieldname
s_row_no STRUCTURE s_roid.

    FIELD-SYMBOLS: <fv_cotiza> TYPE LINE OF gt_cotiza_outtab.

    *   DATA: t_valor TYPE STANDARD TABLE OF ty_valor.
    REFRESH t_valor.
    DATA: w_valor TYPE LINE OF t_valor.
    DATA: t_modelos      TYPE STANDARD TABLE OF ztpp_config WITH HEADER LINE
    ,
    t_modelos2      TYPE STANDARD TABLE OF ztpp_config WITH HEADER LINE
    ,
    t_modelos_aux TYPE STANDARD TABLE OF ztpp_config WITH HEADER LINE
    ,
    lw_modelos      TYPE LINE OF t_modelos,
    lv_contador     TYPE i,
    lw_cotiza       TYPE LINE OF gt_cotiza_outtab,
    lw_cotiza2      TYPE LINE OF gt_cotiza_outtab,
    lv_inserta      TYPE c.
    DATA: dmpv_cant TYPE i,
    lv_cant2 TYPE i,
    lv_index TYPE i.

    DATA: gt_cotiza_outtab_aux TYPE STANDARD TABLE OF gty_cotiza_outtab WITH
    WITH HEADER LINE.
    gv_codigo = dmp200_producto.
    * CALL METHOD grid_cotiza->check_changed_data.

```

```

REFRESH: t_valor,t_modelos,t_modelos_aux,taux_cotizacion,taux_plantilla
.
SELECT * FROM ztpp_config INTO TABLE t_modelos WHERE codigo = gv_codig
o.
gt_cotiza_outtab_aux[] = gt_cotiza_outtab[].

READ TABLE gt_cotiza_outtab INTO lw_cotiza2 INDEX p_es_row_no-row_id.
IF lw_cotiza2-subtitulo <> 'MARCA-MODELO-CHASIS' AND
lw_cotiza2-subtitulo <> 'MARCA-MODELO-MOTOR' AND
lw_cotiza2-subtitulo <> 'COMBUSTIBLE' AND
lw_cotiza2-subtitulo <> 'FORMULA RODANTE'.

.

lv_index = p_es_row_no-row_id.

* DELETE gt_cotiza_outtab_aux WHERE st_fila = 1.
LOOP AT gt_cotiza_outtab_aux WHERE titulo = lw_cotiza2-
titulo AND
subtitulo = lw_cotiza2-
subtitulo AND
caracterica = lw_cotiza2-
caracterica AND
subcaracteristica = lw_cotiza2-
subcaracteristica.
IF sy-subrc = 0.
p_es_row_no-row_id = sy-tabix.
ENDIF.

ENDLOOP.

t_modelos_aux[] = t_modelos[].
t_modelos2[] = t_modelos[].

DELETE ADJACENT DUPLICATES FROM t_modelos_aux COMPARING secuen_config
.

```



```

LOOP AT t_modelos_aux.
  CLEAR: lv_contador, lv_inserta.
  REFRESH: taux_cotizacion,taux_plantilla..
  LOOP AT t_modelos WHERE secuen_config = t_modelos_aux-
secuen_config.
    ADD 1 TO lv_contador.
    MOVE-CORRESPONDING t_modelos TO taux_plantilla.
    APPEND taux_plantilla.

    READ TABLE gt_cotiza_outtab_aux INTO lw_cotiza INDEX lv_contador.
    IF sy-subrc = 0.
*      if lw_cotiza-ST_FILA <> 1.
        MOVE-CORRESPONDING lw_cotiza TO taux_cotizacion.
        APPEND taux_cotizacion.
*      endif.
    ENDIF.

    IF lv_contador = ( p_es_row_no-row_id - 1 ).
      IF taux_cotizacion[] EQ taux_plantilla[].
        lv_inserta = 'X'.
        ADD 1 TO lv_contador.

        READ TABLE t_modelos INTO lw_modelos INDEX p_es_row_no-
row_id.
        IF sy-subrc = 0.
          w_valor-valor = lw_modelos-valor.
          APPEND w_valor TO t_valor.
          DELETE t_modelos WHERE secuen_config = t_modelos_aux-
secuen_config.
            EXIT.
          ENDIF.

        ELSE.
          DELETE t_modelos WHERE secuen_config = t_modelos_aux-
secuen_config.
            EXIT.
          ENDIF.
        ENDIF.
*      if lv_inserta = 'X'.
*
*      ENDIF.
    ENDLOOP.

```

```

ENDLOOP.

SORT t_valor ASCENDING BY valor.
DELETE ADJACENT DUPLICATES FROM t_valor COMPARING valor.
* w_valor-valor = 'aa'.
* APPEND w_valor TO t_valor.
*
* w_valor-valor = 'bb'.
* APPEND w_valor TO t_valor.

CALL FUNCTION 'F4IF_INT_TABLE_VALUE_REQUEST'
EXPORTING
    retfield      = 'MOTOR'
    value_org     = 'S'
TABLES
    value_tab     = t_valor
    return_tab    = return_tab
EXCEPTIONS
    parameter_error = 1
    no_values_found = 2
    OTHERS         = 3.
IF NOT return_tab[] IS INITIAL.
    READ TABLE return_tab INDEX 1.
    READ TABLE gt_cotiza_outtab INDEX lv_index
    ASSIGNING <fv_cotiza>.
*
    <fv_cotiza>-valor = return_tab-fieldval.
    <fv_cotiza>-icon = '@5B@'.
    <fv_cotiza>-st_fil = ''.
*-----
    gt_cotiza_outtab_aux[] = gt_cotiza_outtab[].
*
*
    DELETE gt_cotiza_outtab_aux WHERE st_fil = 1.
    LOOP AT gt_cotiza_outtab_aux WHERE titulo = lw_cotiza2
-titulo AND
                                subtitulo = lw_cotiza2
-subtitulo AND
                                caracterica = lw_cotiza2
-caracterica AND

```

```

subcaracteristica = lw_cotiza2
-subcaracteristica.
    IF sy-subrc = 0.
        p_es_row_no-row_id = sy-tabix.
    ENDIF.

ENDLOOP.
DELETE ADJACENT DUPLICATES FROM t_modelos_aux COMPARING secuen_conf
ig.

REFRESH: taux_cotizacion,taux_plantilla.
LOOP AT t_modelos_aux.
    CLEAR: lv_contador, lv_inserta.
    REFRESH: taux_cotizacion,taux_plantilla..
    LOOP AT t_modelos2 WHERE secuen_config = t_modelos_aux-
secuen_config.
        ADD 1 TO lv_contador.
        MOVE-CORRESPONDING t_modelos2 TO taux_plantilla.
        APPEND taux_plantilla.

        READ TABLE gt_cotiza_outtab_aux INTO lw_cotiza INDEX lv_contado
r.

        IF sy-subrc = 0.
*         if lw_cotiza-ST_FILA <> 1.
            MOVE-CORRESPONDING lw_cotiza TO taux_cotizacion.
            APPEND taux_cotizacion.
*         endif.
        ENDIF.

        IF lv_contador = ( p_es_row_no-row_id ).
            IF taux_cotizacion[] EQ taux_plantilla[].
*                lv_inserta = 'X'.
                ADD 1 TO lv_contador.

                READ TABLE t_modelos2 INTO lw_modelos INDEX p_es_row_no-
row_id.

                IF sy-subrc = 0.
                    w_valor-valor = lw_modelos-valor.
*                    APPEND w_valor TO t_valor.
                    DELETE t_modelos2 WHERE secuen_config <> t_modelos_aux-
secuen_config.

                    EXIT.

```

```

        ENDIF.

    ELSE.
        DELETE t_modelos2 WHERE secuen_config = t_modelos_aux-
secuen_config.
        EXIT.
    ENDIF.
ENDIF.
ENDIF.
*   if lv_inserta = 'X'.
*
*   ENDIF.
ENDLOOP.
ENDLOOP.

LOOP AT t_modelos2 .

    READ TABLE   gt_cotiza_outtab ASSIGNING <fv_cotiza> WITH KEY
titulo           = t_modelos2-titulo

subtitulo        = t_modelos2-subtitulo

caracterica      = t_modelos2-caracterica

subcaracteristica = t_modelos2-subcaracteristica.
    IF sy-subrc = 0.
        <fv_cotiza>-prec_tit = t_modelos2-prec_tit.
    ENDIF.

    IF p_es_row_no-row_id < sy-tabix.
        READ TABLE   gt_cotiza_outtab ASSIGNING <fv_cotiza> WITH KEY
titulo           = t_modelos2-titulo

subtitulo        = t_modelos2-subtitulo

caracterica      = t_modelos2-caracterica

subcaracteristica = t_modelos2-subcaracteristica.
        IF sy-subrc = 0.
            <fv_cotiza>-valor = t_modelos2-valor.
        ELSE.

```

```
        CLEAR
        <fv_cotiza>-valor.
    ENDIF.
ENDIF.
ENDLOOP.
```

```
*-----
```

```
PERFORM procesa_data.
```

```
DATA: is_stable TYPE lvc_s_stbl.
is_stable-row = 'X'.
is_stable-col = 'X'.
```

```
grid_cotiza->refresh_table_display(
    EXPORTING
        is_stable = is_stable ).
```

```
ENDIF.
```

```
ENDIF.
```

```
ENDFORM.
```

```
FORM handle1_mach2 USING p_e_fieldname
s_row_no STRUCTURE s_roid.
```

```
ENDFORM.
```

```
FORM handle1_mach3 USING p_e_fieldname
```

s_row_no STRUCTURE s_roid.

FIELD-SYMBOLS: <fv_cotiza> TYPE LINE OF gt_cotiza_outtab,
 <fs_cal> TYPE any,
 <fs_val> TYPE any,
 <fs_val2> TYPE any.

DATA: dmpv_restriccion TYPE c.

CLEAR lv_restriccion.

* DATA: t_valor TYPE STANDARD TABLE OF ty_valor.

REFRESH t_valor.

DATA: w_valor TYPE LINE OF t_valor,
 w_config TYPE LINE OF t_config,
 lw_cotiza2 TYPE LINE OF gt_cotiza_outtab,
 nombre TYPE string,
 nombre3 TYPE string,
 nombre2 TYPE string

.

DATA: dmpv_cant TYPE n,
 dmpv_cant2 TYPE n,
 dmpv_index TYPE n.

DATA: l_plantillas1 TYPE LINE OF t_plantillas1,
 l_plantillas2 TYPE LINE OF t_plantillas2,
 l_compara.

DATA: t_plantillas1_aux TYPE STANDARD TABLE OF ztpp_valor1,
 t_plantillas2_aux TYPE STANDARD TABLE OF ztpp_valor2.

DATA:lv_contador(3) TYPE n VALUE '000'.

DATA: gt_cotiza_outtab_aux TYPE STANDARD TABLE OF gty_cotiza_outtab WITH
 HEADER LINE.

gv_codigo = dmp200_producto.

* CALL METHOD grid_cotiza->check_changed_data.

REFRESH: t_valor.

TYPES: BEGIN OF ty_mach.

 INCLUDE STRUCTURE ztpp_valor1.

TYPES:col1080(50),

col1081(50),

col1082(50),

col1083(50),

col1084 (50) ,
col1085 (50) ,
col1086 (50) ,
col1087 (50) ,
col1088 (50) ,
col1089 (50) ,
col1090 (50) ,
col1091 (50) ,
col1092 (50) ,
col1093 (50) ,
col1094 (50) ,
col1095 (50) ,
col1096 (50) ,
col1097 (50) ,
col1098 (50) ,
col1099 (50) ,
col1100 (50) ,
col1101 (50) ,
col1102 (50) ,
col1103 (50) ,
col1104 (50) ,
col1105 (50) ,
col1106 (50) ,
col1107 (50) ,
col1108 (50) ,
col1109 (50) ,
col1110 (50) ,
col1111 (50) ,
col1112 (50) ,
col1113 (50) ,
col1114 (50) ,
col1115 (50) ,
col1116 (50) ,
col1117 (50) ,
col1118 (50) ,
col1119 (50) ,
col1120 (50) ,
col1121 (50) ,
col1122 (50) ,
col1123 (50) ,
col1124 (50) ,
col1125 (50) ,

```
col126(50),
col127(50),
col128(50),
col129(50),
col130(50),
col131(50),
col132(50),
col133(50),
col134(50),
col135(50),
col136(50),
col137(50),
col138(50),
col139(50),
col140(50),
```

```
END OF ty_mach.
```

```
DATA: lt_configu_2 TYPE STANDARD TABLE OF ty_restr WITH HEADER LINE.
DATA: lt_rest1 TYPE STANDARD TABLE OF ztppl_restr1 WITH HEADER LINE.
DATA: lt_rest2 TYPE STANDARD TABLE OF ztppl_restr2 WITH HEADER LINE.
DATA: wa_configu_2 TYPE ty_restr .
DATA: wa_configu_2_aux TYPE ty_restr .
FIELD-SYMBOLS: <fs_configu_2> TYPE LINE OF lt_configu_2.
DATA: wa_mach TYPE ty_mach .
```

```
DATA: t_mach TYPE STANDARD TABLE OF ty_mach WITH HEADER LINE.
```

```
lt_rest1[] = t_rest1[].
```

```
lt_rest2[] = t_rest2[].
```

```
lt_configu_2[] = t_configu_2[].
```

```
LOOP AT lt_rest1.
```

```
  CLEAR: l_compara,lv_contador.
```

```
  READ TABLE lt_rest2 WITH KEY fila = lt_rest1-fila BINARY SEARCH.
```

```
  MOVE-CORRESPONDING lt_rest1 TO t_mach.
```

```
  MOVE-CORRESPONDING lt_rest2 TO t_mach.
```

```
  APPEND t_mach.
```

```
ENDLOOP.
```

```
LOOP AT lt_configu_2 ASSIGNING <fs_configu_2>.
```

```
  READ TABLE t_config WITH KEY titulo = <fs_configu_2>-
```



```

titulo subtit = <fs_configu_2>-subtit
                                caract = <fs_configu_2>-
caract subcar = <fs_configu_2>-subcar.
  IF sy-subrc = 0.
    <fs_configu_2>-columna_val = t_config-columna.
  ENDIF.
ENDLOOP.

"ADD DB7 VISUALIZAR CARACT EN COTIZAC INI
READ TABLE gt_cotiza_outtab_alv_x INDEX p_es_row_no-row_id.
IF sy-subrc = 0.
  p_es_row_no-row_id = gt_cotiza_outtab_alv_x-ord_tit.
ENDIF.

"ADD DB7 VISUALIZAR CARACT EN COTIZAC END
READ TABLE gt_cotiza_outtab INTO lw_cotiza2 INDEX p_es_row_no-row_id.
IF lw_cotiza2-subtitulo <> 'MARCA-MODELO-CHASIS' AND
   lw_cotiza2-subtitulo <> 'MARCA-MODELO-MOTOR' AND
   lw_cotiza2-subtitulo <> 'COMBUSTIBLE' AND
   lw_cotiza2-subtitulo <> 'FORMULA RODANTE'.

  READ TABLE t_config INTO w_config WITH KEY titulo = lw_cotiza2-
titulo
                                                subtit = lw_cotiza2-
subtitulo
                                                caract = lw_cotiza2-
caracterica
                                                subcar = lw_cotiza2-
subcaracteristica.

  IF sy-subrc = 0.
    "CONSULTA SI HAY ALGUNA RESTRICION
    SORT lt_configu_2 BY columna_val ASCENDING.
    LOOP AT lt_configu_2 INTO wa_configu_2 WHERE columna_val <= w_confi
g-columna.
    ENDLOOP.
    SORT lt_configu_2 BY columna ASCENDING.
    IF wa_configu_2 IS NOT INITIAL.
      DATA: colum_rest(3) TYPE n.
      LOOP AT lt_configu_2 INTO wa_configu_2_aux.
        ADD 1 TO colum_rest.
      *****OBTENER VALOR DE COLUMNA DE LA PLANTILLA*****

```

```

*****
      IF wa_configu_2_aux-columna_val < 80.
          CONCATENATE 'wa_plantilla1-col' wa_configu_2_aux-
columna_val INTO nombre.
      ELSE.
          CONCATENATE 'wa_plantilla2-col' wa_configu_2_aux-
columna_val INTO nombre.
      ENDIF.
      ASSIGN (nombre) TO <fs_val>.
*****OBTENER VALOR DE COLUMNA DE LA PLANTILLA*****
*****

*****ELIMINAR RESTRICCIONES SEGUN COLUMNA*****
      IF wa_configu_2_aux-columna_val <> w_config-columna.
          CASE wa_configu_2_aux-columna.
              WHEN '001'.
                  DELETE t_mach WHERE col001 <> <fs_val>.
              WHEN '002'.
                  DELETE t_mach WHERE col002 <> <fs_val>.
              WHEN '003'.
                  DELETE t_mach WHERE col003 <> <fs_val>.
              WHEN '004'.
                  DELETE t_mach WHERE col004 <> <fs_val>.
              WHEN '005'.
                  DELETE t_mach WHERE col005 <> <fs_val>.
              WHEN '006'. DELETE t_mach WHERE col006 <> <fs_val>.
              WHEN '007'. DELETE t_mach WHERE col007 <> <fs_val>.
              WHEN '008'. DELETE t_mach WHERE col008 <> <fs_val>.
              WHEN '009'. DELETE t_mach WHERE col009 <> <fs_val>.
              WHEN '010'. DELETE t_mach WHERE col010 <> <fs_val>.
              WHEN '011'. DELETE t_mach WHERE col011 <> <fs_val>.
              WHEN '012'. DELETE t_mach WHERE col012 <> <fs_val>.
              WHEN '013'. DELETE t_mach WHERE col013 <> <fs_val>.
              WHEN '014'. DELETE t_mach WHERE col014 <> <fs_val>.
              WHEN '015'. DELETE t_mach WHERE col015 <> <fs_val>.
              WHEN '016'. DELETE t_mach WHERE col016 <> <fs_val>.
              WHEN '017'. DELETE t_mach WHERE col017 <> <fs_val>.
              WHEN '018'. DELETE t_mach WHERE col018 <> <fs_val>.
              WHEN '019'. DELETE t_mach WHERE col019 <> <fs_val>.
              WHEN '020'. DELETE t_mach WHERE col020 <> <fs_val>.
              WHEN '021'. DELETE t_mach WHERE col021 <> <fs_val>.
              WHEN '022'. DELETE t_mach WHERE col022 <> <fs_val>.
          END CASE.
      END IF.

```

WHEN '023'. DELETE t_mach WHERE col023 <> <fs_val>.
WHEN '024'. DELETE t_mach WHERE col024 <> <fs_val>.
WHEN '025'. DELETE t_mach WHERE col025 <> <fs_val>.
WHEN '026'. DELETE t_mach WHERE col026 <> <fs_val>.
WHEN '027'. DELETE t_mach WHERE col027 <> <fs_val>.
WHEN '028'. DELETE t_mach WHERE col028 <> <fs_val>.
WHEN '029'. DELETE t_mach WHERE col029 <> <fs_val>.
WHEN '030'. DELETE t_mach WHERE col030 <> <fs_val>.
WHEN '031'. DELETE t_mach WHERE col031 <> <fs_val>.
WHEN '032'. DELETE t_mach WHERE col032 <> <fs_val>.
WHEN '033'. DELETE t_mach WHERE col033 <> <fs_val>.
WHEN '034'. DELETE t_mach WHERE col034 <> <fs_val>.
WHEN '035'. DELETE t_mach WHERE col035 <> <fs_val>.
WHEN '036'. DELETE t_mach WHERE col036 <> <fs_val>.
WHEN '037'. DELETE t_mach WHERE col037 <> <fs_val>.
WHEN '038'. DELETE t_mach WHERE col038 <> <fs_val>.
WHEN '039'. DELETE t_mach WHERE col039 <> <fs_val>.
WHEN '040'. DELETE t_mach WHERE col040 <> <fs_val>.
WHEN '041'. DELETE t_mach WHERE col041 <> <fs_val>.
WHEN '042'. DELETE t_mach WHERE col042 <> <fs_val>.
WHEN '043'. DELETE t_mach WHERE col043 <> <fs_val>.
WHEN '044'. DELETE t_mach WHERE col044 <> <fs_val>.
WHEN '045'. DELETE t_mach WHERE col045 <> <fs_val>.
WHEN '046'. DELETE t_mach WHERE col046 <> <fs_val>.
WHEN '047'. DELETE t_mach WHERE col047 <> <fs_val>.
WHEN '048'. DELETE t_mach WHERE col048 <> <fs_val>.
WHEN '049'. DELETE t_mach WHERE col049 <> <fs_val>.
WHEN '050'. DELETE t_mach WHERE col050 <> <fs_val>.
WHEN '051'. DELETE t_mach WHERE col051 <> <fs_val>.
WHEN '052'. DELETE t_mach WHERE col052 <> <fs_val>.
WHEN '053'. DELETE t_mach WHERE col053 <> <fs_val>.
WHEN '054'. DELETE t_mach WHERE col054 <> <fs_val>.
WHEN '055'. DELETE t_mach WHERE col055 <> <fs_val>.
WHEN '056'. DELETE t_mach WHERE col056 <> <fs_val>.
WHEN '057'. DELETE t_mach WHERE col057 <> <fs_val>.
WHEN '058'. DELETE t_mach WHERE col058 <> <fs_val>.
WHEN '059'. DELETE t_mach WHERE col059 <> <fs_val>.
WHEN '060'. DELETE t_mach WHERE col060 <> <fs_val>.
WHEN '061'. DELETE t_mach WHERE col061 <> <fs_val>.
WHEN '062'. DELETE t_mach WHERE col062 <> <fs_val>.
WHEN '063'. DELETE t_mach WHERE col063 <> <fs_val>.
WHEN '064'. DELETE t_mach WHERE col064 <> <fs_val>.

WHEN '065'. DELETE t_mach WHERE col065 <> <fs_val>.
WHEN '066'. DELETE t_mach WHERE col066 <> <fs_val>.
WHEN '067'. DELETE t_mach WHERE col067 <> <fs_val>.
WHEN '068'. DELETE t_mach WHERE col068 <> <fs_val>.
WHEN '069'. DELETE t_mach WHERE col069 <> <fs_val>.
WHEN '070'. DELETE t_mach WHERE col070 <> <fs_val>.
WHEN '071'. DELETE t_mach WHERE col071 <> <fs_val>.
WHEN '072'. DELETE t_mach WHERE col072 <> <fs_val>.
WHEN '073'. DELETE t_mach WHERE col073 <> <fs_val>.
WHEN '074'. DELETE t_mach WHERE col074 <> <fs_val>.
WHEN '075'. DELETE t_mach WHERE col075 <> <fs_val>.
WHEN '076'. DELETE t_mach WHERE col076 <> <fs_val>.
WHEN '077'. DELETE t_mach WHERE col077 <> <fs_val>.
WHEN '078'. DELETE t_mach WHERE col078 <> <fs_val>.
WHEN '079'. DELETE t_mach WHERE col079 <> <fs_val>.
WHEN '080'. DELETE t_mach WHERE col080 <> <fs_val>.
WHEN '081'. DELETE t_mach WHERE col081 <> <fs_val>.
WHEN '082'. DELETE t_mach WHERE col082 <> <fs_val>.
WHEN '083'. DELETE t_mach WHERE col083 <> <fs_val>.
WHEN '084'. DELETE t_mach WHERE col084 <> <fs_val>.
WHEN '085'. DELETE t_mach WHERE col085 <> <fs_val>.
WHEN '086'. DELETE t_mach WHERE col086 <> <fs_val>.
WHEN '087'. DELETE t_mach WHERE col087 <> <fs_val>.
WHEN '088'. DELETE t_mach WHERE col088 <> <fs_val>.
WHEN '089'. DELETE t_mach WHERE col089 <> <fs_val>.
WHEN '090'. DELETE t_mach WHERE col090 <> <fs_val>.
WHEN '091'. DELETE t_mach WHERE col091 <> <fs_val>.
WHEN '092'. DELETE t_mach WHERE col092 <> <fs_val>.
WHEN '093'. DELETE t_mach WHERE col093 <> <fs_val>.
WHEN '094'. DELETE t_mach WHERE col094 <> <fs_val>.
WHEN '095'. DELETE t_mach WHERE col095 <> <fs_val>.
WHEN '096'. DELETE t_mach WHERE col096 <> <fs_val>.
WHEN '097'. DELETE t_mach WHERE col097 <> <fs_val>.
WHEN '098'. DELETE t_mach WHERE col098 <> <fs_val>.
WHEN '099'. DELETE t_mach WHERE col099 <> <fs_val>.
WHEN '100'. DELETE t_mach WHERE col100 <> <fs_val>.
WHEN '101'. DELETE t_mach WHERE col101 <> <fs_val>.
WHEN '102'. DELETE t_mach WHERE col102 <> <fs_val>.
WHEN '103'. DELETE t_mach WHERE col103 <> <fs_val>.
WHEN '104'. DELETE t_mach WHERE col104 <> <fs_val>.
WHEN '105'. DELETE t_mach WHERE col105 <> <fs_val>.
WHEN '106'. DELETE t_mach WHERE col106 <> <fs_val>.

```
WHEN '107'. DELETE t_mach WHERE col107 <> <fs_val>.
WHEN '108'. DELETE t_mach WHERE col108 <> <fs_val>.
WHEN '109'. DELETE t_mach WHERE col109 <> <fs_val>.
WHEN '110'. DELETE t_mach WHERE col110 <> <fs_val>.
WHEN '111'. DELETE t_mach WHERE col111 <> <fs_val>.
WHEN '112'. DELETE t_mach WHERE col112 <> <fs_val>.
WHEN '113'. DELETE t_mach WHERE col113 <> <fs_val>.
WHEN '114'. DELETE t_mach WHERE col114 <> <fs_val>.
WHEN '115'. DELETE t_mach WHERE col115 <> <fs_val>.
WHEN '116'. DELETE t_mach WHERE col116 <> <fs_val>.
WHEN '117'. DELETE t_mach WHERE col117 <> <fs_val>.
WHEN '118'. DELETE t_mach WHERE col118 <> <fs_val>.
WHEN '119'. DELETE t_mach WHERE col119 <> <fs_val>.
WHEN '120'. DELETE t_mach WHERE col120 <> <fs_val>.
WHEN '121'. DELETE t_mach WHERE col121 <> <fs_val>.
WHEN '122'. DELETE t_mach WHERE col122 <> <fs_val>.
WHEN '123'. DELETE t_mach WHERE col123 <> <fs_val>.
WHEN '124'. DELETE t_mach WHERE col124 <> <fs_val>.
WHEN '125'. DELETE t_mach WHERE col125 <> <fs_val>.
WHEN '126'. DELETE t_mach WHERE col126 <> <fs_val>.
WHEN '127'. DELETE t_mach WHERE col127 <> <fs_val>.
WHEN '128'. DELETE t_mach WHERE col128 <> <fs_val>.
WHEN '129'. DELETE t_mach WHERE col129 <> <fs_val>.
WHEN '130'. DELETE t_mach WHERE col130 <> <fs_val>.
WHEN '131'. DELETE t_mach WHERE col131 <> <fs_val>.
WHEN '132'. DELETE t_mach WHERE col132 <> <fs_val>.
WHEN '133'. DELETE t_mach WHERE col133 <> <fs_val>.
WHEN '134'. DELETE t_mach WHERE col134 <> <fs_val>.
WHEN '135'. DELETE t_mach WHERE col135 <> <fs_val>.
WHEN '136'. DELETE t_mach WHERE col136 <> <fs_val>.
WHEN '137'. DELETE t_mach WHERE col137 <> <fs_val>.
WHEN '138'. DELETE t_mach WHERE col138 <> <fs_val>.
WHEN '139'. DELETE t_mach WHERE col139 <> <fs_val>.
WHEN '140'. DELETE t_mach WHERE col140 <> <fs_val>.
```

```
ENDCASE.
```

```
ENDIF.
```

```
*****ELIMINAR RESTRICCIONES SEGUN COLUMNA*****
```

```
IF colum_rest >= wa_configu_2-columna.
```

```

EXIT.
ENDIF.
ENDLOOP.

```

```

*          DO wa_configu_2-columna TIMES.
*          ADD 1 TO colum_rest.
*
*          ENDDO.

```

```

IF wa_configu_2-columna_val = w_config-columna.

```

```

  LOOP AT t_mach .

```

```

    CONCATENATE 't_mach-col' wa_configu_2-columna INTO nombre .
    ASSIGN (nombre) TO <fs_val>.

```

```

    w_valor-valor = <fs_val>.

```

```

    w_valor-fila = t_mach-fila.

```

```

    APPEND w_valor TO t_valor.

```

```

    CLEAR w_valor.

```

```

  ENDLOOP.

```

```

  lv_restriccion = 'X'.

```

```

ELSE.

```

```

  LOOP AT t_vc WHERE titulo = w_config-

```

```

  titulo AND subtitulo = w_config-subtit AND caracterica = w_config-
  caract AND subcaracteristica = w_config-subcar.

```

```

    w_valor-valor = t_vc-valor.

```

```

    w_valor-fila = t_vc-ord_tit.

```

```

    APPEND w_valor TO t_valor.

```

```

    CLEAR w_valor.

```

```

  ENDLOOP.

```

```

ENDIF.

```

```

ELSE.

```

```

  LOOP AT t_vc WHERE titulo = w_config-

```

```

  titulo AND subtitulo = w_config-subtit AND caracterica = w_config-
  caract AND subcaracteristica = w_config-subcar.

```

```

    w_valor-valor = t_vc-valor.

```

```

        w_valor-fila =      t_vc-ord_tit.
        APPEND w_valor TO t_valor.
        CLEAR w_valor.
    ENDLOOP.

ENDIF.

IF t_mach[] IS NOT INITIAL.
    READ TABLE t_mach INTO wa_mach INDEX 1.
ENDIF.

*****
*
SORT t_valor  ASCENDING BY valor fila ASCENDING.
DELETE ADJACENT DUPLICATES FROM t_valor COMPARING valor.

IF t_valor[] IS NOT INITIAL.
    DATA: field_tab TYPE STANDARD TABLE OF dfies WITH HEADER LINE.
    DATA: dynpfld_mapping TYPE STANDARD TABLE OF dselc WITH HEADER LI
NE.

    CALL FUNCTION 'F4IF_INT_TABLE_VALUE_REQUEST'
        EXPORTING
            retfield      = 'VALOR'
            value_org     = 'S'
        TABLES
            value_tab     = t_valor
            return_tab    = return_tab
*           FIELD_TAB    = FIELD_TAB
*           DYNPFLD_MAPPING = DYNPFLD_MAPPING
    EXCEPTIONS
        parameter_error = 1
        no_values_found = 2
        OTHERS          = 3.

    IF NOT return_tab[] IS INITIAL.
        READ TABLE return_tab INDEX 1.
        READ TABLE gt_cotiza_outtab INDEX p_es_row_no-row_id
        ASSIGNING <fv_cotiza>.

*
*****
        DATA: dmpv_nom_campo TYPE string.
        FIELD-SYMBOLS: <fs> TYPE any.

        IF w_config-columna < 80 .

```

```

        lv_nom_campo = 'wa_plantilla1-col'.
ELSE.
        lv_nom_campo = 'wa_plantilla2-col'.
ENDIF.
CONCATENATE lv_nom_campo w_config-columna INTO lv_nom_campo.
CONDENSE lv_nom_campo.
ASSIGN (lv_nom_campo) TO <fs>.
<fs> = return_tab-fieldval.
* *****
        <fv_cotiza>-valor = return_tab-fieldval.
        <fv_cotiza>-icon = '@5B@'.
        <fv_cotiza>-st_fila = ''.
        IF lv_restriccion = 'X'.
            READ TABLE t_valor INTO w_valor WITH KEY valor = return_tab-
fieldval.

            READ TABLE t_mach INTO wa_mach WITH KEY fila = w_valor-fila.
            LOOP AT t_mach.
                CONCATENATE 'wa_mach-col' wa_configu_2-columna INTO nombre.
                ASSIGN (nombre) TO <fs_val2>.
                CONCATENATE 't_mach-col' wa_configu_2-columna INTO nombre.
                ASSIGN (nombre) TO <fs_val>.
                IF <fs_val> <> <fs_val2>.
                    DELETE t_mach INDEX sy-tabix.
                ENDIF.
            ENDLOOP.
        ELSE.
            READ TABLE t_mach INTO wa_mach INDEX 1.
        ENDIF.

DATA: cont(3) TYPE n,
      colum(3) TYPE n.
cont = w_config-columna.
colum = 140 - cont.

DO colum TIMES.
    ADD 1 TO cont.
    IF cont < 80.
        CONCATENATE 'wa_plantilla1-col' cont INTO nombre.
        ASSIGN (nombre) TO <fs_val>.
    ELSE.
        CONCATENATE 'wa_plantilla2-col' cont INTO nombre.

```



```

        ASSIGN (nombre) TO <fs_val>.
    ENDIF.

    READ TABLE lt_configu_2 INTO wa_configu_2 WITH KEY columna_va
1 = cont.
    IF sy-subrc = 0.
        CONCATENATE 'wa_mach-col' wa_configu_2-columna INTO nombre.
        ASSIGN (nombre) TO <fs_val2>.
        IF <fs_val2> IS NOT INITIAL.
            <fs_val> = <fs_val2>.
        ENDIF.
    ELSE.
        READ TABLE t_config WITH KEY columna = cont.
        IF sy-subrc = 0.
            READ TABLE t_vc WITH KEY titulo = t_config-
titulo subtitulo = t_config-subtit
                                                    caracterica = t_config-
caract subcaracteristica = t_config-subcar.
            IF sy-subrc = 0.
                <fs_val> = t_vc-valor.
            ENDIF.
        ENDIF.
    ENDIF.

ENDDO.

cont = w_config-columna - 1 .
DO colum TIMES.
    ADD 1 TO cont.
    IF cont < 80.
        CONCATENATE 'wa_flag1-col' cont INTO nombre.
        ASSIGN (nombre) TO <fs_val>.
        <fs_val> = 'X'.

    ELSE.
        CONCATENATE 'wa_flag2-col' cont INTO nombre.
        ASSIGN (nombre) TO <fs_val>.
        <fs_val> = 'X'.

    ENDIF.
ENDDO.

```

```

PERFORM recalcula_precios.

PERFORM formatear_plantilla TABLES gt_cotiza_outtab.

PERFORM procesa_data.

DATA: is_stable TYPE lvc_s_stbl.
is_stable-row = 'X'.
is_stable-col = 'X'.
).

ENDIF.

```

```

*F4_ALV
*&-----*

```

```

FORM f4_alv.
* DATA: event_receiver TYPE REF TO lcl_event_receiver.
* data: wa_f4 TYPE LINE OF lt_f4.
REFRESH lt_f4.

```

```

lt_f4-fieldname = 'VALOR'.
lt_f4-register = 'X' .
lt_f4-getbefore = 'X' .

```

```

lt_f4-chngeafter = 'X' .
APPEND lt_f4.

* CREATE OBJECT event_receiver.
* SET HANDLER event_receiver->handle_data_changed FOR dmp100c_alv1.

* ENDIF.
ENDFORM.

*&-----*
*&      Form  GENERA_MATCHODE
*&-----*

FORM genera_matchode USING VALUE (p_0078)
                           VALUE (p_0079) .      CASE p_0078.

  WHEN: 'TIP_VTA'.
    MOVE t_match-fieldval TO p_tvta.
    CLEAR: p_ano,p_mbus,p_tcar,p_vers,p_macha,p_mocha,p_comb,p_mot,p_
m_mot.

  WHEN: 'ANO'.
    MOVE t_match-fieldval TO p_ano.
    CLEAR: p_mbus,p_tcar,p_vers,p_macha,p_mocha,p_comb,p_mot,p_m_mot.

  WHEN: 'MOD_BUS'.
    MOVE t_match-fieldval TO p_mbus.
    CLEAR: p_tcar,p_vers,p_macha,p_mocha,p_comb,p_mot,p_m_mot.

  WHEN: 'TIP_CAR'.
    MOVE t_match-fieldval TO p_tcar.
    CLEAR: p_vers,p_macha,p_mocha,p_comb,p_mot,p_m_mot.

  WHEN: 'VERS'.
    MOVE t_match-fieldval TO p_vers.
    CLEAR: p_macha,p_mocha,p_comb,p_mot,p_m_mot.

  WHEN: 'MAR_CHA'.
    MOVE t_match-fieldval TO p_macha.
    CLEAR: p_mocha,p_comb,p_mot,p_m_mot.

  WHEN: 'MOD_CHA'.
    MOVE t_match-fieldval TO p_mocha.
    CLEAR: p_comb,p_mot,p_m_mot.

  WHEN: 'COBUST'.
    MOVE t_match-fieldval TO p_comb.
    CLEAR: p_mot,p_m_mot.

```

```

    WHEN: 'MOTOR'.
        MOVE t_match-fieldval TO p_mot.
        CLEAR: p_m_mot.
    WHEN: 'MOD_MOTOR'.
        MOVE t_match-fieldval TO p_m_mot.
    ENDCASE.
ENDIF.

DATA : t_name          TYPE TABLE OF dynpread,
      wls_dyfield TYPE dynpread.

REFRESH t_name.
CLEAR ls_dynfield.
ls_dynfield-fieldname = 'VP_TVTA'.
ls_dynfield-fieldvalue = p_tvta.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_ANO'.
ls_dynfield-fieldvalue = p_ano.
IF p_ano IS INITIAL.
    ls_dynfield-fieldvalue = ''.
ELSE.
    ls_dynfield-fieldvalue = p_ano.
ENDIF.

APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_MBUS'.
ls_dynfield-fieldvalue = p_mbus.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_TCAR'.
ls_dynfield-fieldvalue = p_tcar.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_VERS'.
ls_dynfield-fieldvalue = p_vers.

```

```

APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_MACHA'.
ls_dynfield-fieldvalue = p_macha.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_MOCHA'.
ls_dynfield-fieldvalue = p_mocha.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_COMB'.
ls_dynfield-fieldvalue = p_comb.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_MOT'.
ls_dynfield-fieldvalue = p_mot.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

ls_dynfield-fieldname = 'VP_M_MOT'.
ls_dynfield-fieldvalue = p_m_mot.
APPEND wls_dyfield TO t_name.
CLEAR ls_dynfield.

CALL FUNCTION 'DYNP_VALUES_UPDATE'
  EXPORTING
    dymame      = sy-repid
    dynumb      = sy-dynnr
  TABLES
    dynpfields = t_name.
ENDFORM.

*&-----*
*&      Form  OBTIENE_MATCH
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text

```

```

*-----*
FORM obtiene_match USING VALUE (p_campo) .
REFRESH t_valores.
CASE p_campo.
  WHEN: 'TIP_VTA'.
    LOOP AT gt_claves.
      t_valores-valor = gt_claves-tip_vta.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'ANO'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta .
      t_valores-valor = gt_claves-ano.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'MOD_BUS'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano.
      t_valores-valor = gt_claves-mod_bus.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'TIP_CAR'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus.
      t_valores-valor = gt_claves-tip_car.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'VERS'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND
      tip_car = p_tcar.
      t_valores-valor = gt_claves-vers.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'MAR_CHA'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND
      tip_car = p_tcar AND vers = p_vers.
      t_valores-valor = gt_claves-mar_cha.
      APPEND t_valores.
    ENDLOOP.
  WHEN: 'MOD_CHA'.
    LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND

```

```

tip_car = p_tcar AND vers = p_vers AND mar_
cha = p_macha.
t_valores-valor = gt_claves-mod_cha.
APPEND t_valores.
ENDLOOP.
WHEN: 'COBUST'.
LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND
tip_car = p_tcar AND vers = p_vers AND mar_
cha = p_macha AND
mod_cha = p_mocha.
t_valores-valor = gt_claves-cobust.
APPEND t_valores.
ENDLOOP.
WHEN: 'MOTOR'.
LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND
tip_car = p_tcar AND vers = p_vers AND mar_
cha = p_macha AND
mod_cha = p_mocha AND cobust = p_comb.
t_valores-valor = gt_claves-motor.
APPEND t_valores.
ENDLOOP.
WHEN: 'MOD_MOTOR'.
LOOP AT gt_claves WHERE tip_vta = p_tvta AND ano = p_ano AND mod_bu
s = p_mbus AND
tip_car = p_tcar AND vers = p_vers AND mar_
cha = p_macha AND
mod_cha = p_mocha AND cobust = p_comb AND m
otor = p_mot.
t_valores-valor = gt_claves-mod_motor.
APPEND t_valores.
ENDLOOP.
ENDCASE.
ENDFORM.
*&-----*
*& Form MOSTRAR_COTIZACION
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text

```

```
FORM obtner_plantilla_visual USING codigo.
  DATA: lt_config_aux TYPE STANDARD TABLE OF ztpc_configura WITH HEADER L
INE.
  DATA: lwa_config TYPE LINE OF t_config.
  REFRESH: t_plantillas1,t_plantillas2.
  t_plantillas1 FROM ztpc_valor1 WHERE codigo = codigo.
  t_plantillas2 FROM ztpc_valor2 WHERE codigo = codigo.
  t_config FROM ztpc_configura WHERE codigo = codigo.
  lt_config_aux[] = t_config[].
  SORT lt_config_aux BY columna ASCENDING titulo ASCENDING.
  CLEAR lwa_config.
  LOOP AT t_config INTO lwa_config.
  ENDLOOP.
  ADD 1 TO lwa_config-columna.
  lwa_config-titulo = '/'.
  lwa_config-subtit = '/'.
  lwa_config-caract = '/'.
  lwa_config-subcar = '/'.
  APPEND lwa_config TO t_config.
  CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
  LOOP AT lt_config_aux.
  ADD 1 TO lwa_config-columna.
  lwa_config-titulo = lt_config_aux-titulo.
  APPEND lwa_config TO t_config.
  ENDLOOP.
  ADD 1 TO lwa_config-columna.
  lwa_config-titulo = '/'.
  lwa_config-subtit = '/'.
  lwa_config-caract = '/'.
  lwa_config-subcar = '/'.
  APPEND lwa_config TO t_config.
  ADD 1 TO lwa_config-columna.
  lwa_config-titulo = 'TEXT0'.
  CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
  APPEND lwa_config TO t_config.
  ADD 1 TO lwa_config-columna.
  lwa_config-titulo = '/'.
  lwa_config-subtit = '/'.
  lwa_config-caract = '/'.
  lwa_config-subcar = '/'.
```



```

APPEND lwa_config TO t_config.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = 'ESTANDAR'.
CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
APPEND lwa_config TO t_config.

READ TABLE gt_plantillas_st INTO wa_output WITH KEY codigo = dmp200_pr
oducto.
IF sy-subrc = 0.
    p_tvta = wa_output-tip_vta.
    p_ano = wa_output-ano.
    p_mbus = wa_output-mod_bus.
    p_tcar = wa_output-tip_car.
    p_vers = wa_output-vers.
    p_macha = wa_output-mar_cha.
    p_mocha = wa_output-mod_cha.
    p_comb = wa_output-cobust.
    p_mot = wa_output-motor.
    p_m_mot = wa_output-mod_motor.
    p_f_rod = wa_output-form_rod.

    dmp200_macha = p_macha.
    dmp200_mocha = p_mocha.
    dmp200_comb = p_comb .
    dmp200_mot = p_mot .
    dmp200_m_mot = p_m_mot.
    dmp200_f_rod = p_f_rod.

ENDIF.

PERFORM busca_mach USING p_tvta p_mbus p_tcar p_macha p_mocha
                        p_comb p_mot p_m_mot p_f_rod ..

ENDFORM.
FORM obtener_plantilla USING codigo.

DATA: nombre TYPE string.
DATA: monto TYPE string.
DATA: precio_x_cantidad TYPE wrbtr.
FIELD-SYMBOLS: <fs_val> TYPE any.
FIELD-SYMBOLS: <valor> TYPE any.

```

```

DATA: cont(3) TYPE n VALUE '000'.
DATA: cont_prec(3) TYPE n.
TYPES: BEGIN OF tey_pr_tit,
        titulo TYPE ztpp_configura-titulo,
        precio TYPE ze_pp_pv_netto,
      END OF tey_pr_tit.
DATA: let_pr_tit TYPE STANDARD TABLE OF tey_pr_tit WITH HEADER LINE.
FIELD-SYMBOLS: <fs_pr_tit> TYPE LINE OF lt_pr_tit.

REFRESH: t_plantillas1,t_plantillas2,t_config,gt_cotiza_outtab.
REFRESH: t_rest1,t_rest2,t_configu_2,t_vc.
CLEAR: wa_plantilla1, wa_plantilla2,wa_config,gt_cotiza_outtab.
DATA: lwa_config TYPE LINE OF t_config.
DATA: lt_config_aux TYPE STANDARD TABLE OF ztpp_configura WITH HEADER L
INE.

*****

SELECT * FROM ztpp_restr1 INTO TABLE t_rest1 WHERE codigo = codigo.
SELECT * FROM ztpp_restr2 INTO TABLE t_rest2 WHERE codigo = codigo.
SELECT * FROM ztpp_configura2 INTO CORRESPONDING FIELDS OF TABLE t_con
figu_2 WHERE codigo = codigo.
SELECT * FROM ztpp_valor_carac INTO TABLE t_vc WHERE codigo = codigo
AND valor NE space.
SELECT * FROM ztpp_valor_carac INTO TABLE t_vc2 WHERE codigo = codigo
AND valor EQ space.
SELECT * INTO TABLE t_config FROM ztpp_configura WHERE codigo =
codigo.
lt_config_aux[] = t_config[].
SORT lt_config_aux BY columna ASCENDING titulo ASCENDING.
DELETE ADJACENT DUPLICATES FROM lt_config_aux[] COMPARING titulo.
SORT lt_config_aux BY columna ASCENDING.
LOOP AT lt_config_aux.
    MOVE-CORRESPONDING lt_config_aux TO lt_pr_tit.
    APPEND lt_pr_tit.
ENDLOOP.
SELECT * FROM ztpp_standard INTO TABLE t_standard WHERE codigo = codigo
.
IF sy-subrc = 0.
    t_standard INTO 1wa_standard.
    t_config INTO lwa_config WITH KEY titulo = wa_standard-
titulo subtit = wa_standard-subtitulo
                                caract = wa_standard

```

```

-caracteristica subcar = wa_standard-subcaracteristica.
  IF sy-subrc = 0.
    IF lwa_config-columna < 80.
      CONCATENATE 'WA_PLANTILLA1-COL' lwa_config-columna INTO nombre.
    ELSE.
      CONCATENATE 'WA_PLANTILLA2-COL' lwa_config-columna INTO nombre.
    ENDIF.
    ASSIGN (nombre) TO <valor>.
    <valor> = wa_standard-valor.
    cont_prec = lwa_config-columna.
  ENDIF.

  LOOP AT t_vc WHERE titulo = wa_standard-
titulo AND subtitulo = wa_standard-subtitulo AND
                                caracterica = wa_standard-
caracteristica AND subcaracteristica = wa_standard-subcaracteristica
                                AND valor = wa_standard-valor.
    READ TABLE lt_pr_tit ASSIGNING <fs_pr_tit> WITH KEY titulo = t_v
c-titulo.
    IF sy-subrc = 0.
      IF t_vc-cantidad IS NOT INITIAL.
        precio_x_cantidad = t_vc-costo * t_vc-cantidad.
      ELSE.
        precio_x_cantidad = t_vc-costo.
      ENDIF.
      ADD precio_x_cantidad TO <fs_pr_tit>-precio.
    ENDIF.
  ENDLOOP.

*   READ TABLE t_vc WITH KEY titulo = wa_standard-
titulo subtitulo = wa_standard-subtitulo
*
*                                     caracterica = wa_standard-
caracteristica subcaracteristica = wa_standard-
subcaracteristica valor = wa_standard-valor.
*   IF sy-subrc = 0.
*     READ TABLE lt_pr_tit ASSIGNING <fs_pr_tit> WITH KEY titulo = t_
vc-titulo.
*     IF sy-subrc = 0.
*       if t_vc-cantidad
*         precio_x_cantidad = t_vc-costo * t_vc-cantidad.
*       else.
*         precio_x_cantidad = t_vc-costo.
*     endif.

```

```

*          ADD precio_x_cantidad TO <fs_pr_tit>-precio.
*      endif.
*      ENDIF.
ENDLOOP.
LOOP AT t_vc2.
    READ TABLE lt_pr_tit ASSIGNING <fs_pr_tit> WITH KEY titulo = t_vc2
-titulo.
    IF sy-subrc = 0.
        IF t_vc2-cantidad IS NOT INITIAL.
            precio_x_cantidad = t_vc2-costo * t_vc2-cantidad.
        ELSE.
            precio_x_cantidad = t_vc2-costo.
        ENDIF.
        ADD precio_x_cantidad TO <fs_pr_tit>-precio.
    ENDIF.
ENDLOOP.
ADD 1 TO cont_prec.
IF cont_prec < 80.
    CONCATENATE 'WA_PLANTILLA1-COL' cont_prec INTO nombre.
ELSE.
    CONCATENATE 'WA_PLANTILLA2-COL' cont_prec INTO nombre.
ENDIF.
ASSIGN (nombre) TO <valor>.
<valor> = '/'.

LOOP AT lt_pr_tit ASSIGNING <fs_pr_tit>.
    ADD 1 TO cont_prec.
    IF cont_prec < 80.
        CONCATENATE 'WA_PLANTILLA1-COL' cont_prec INTO nombre.
    ELSE.
        CONCATENATE 'WA_PLANTILLA2-COL' cont_prec INTO nombre.
    ENDIF.
    ASSIGN (nombre) TO <valor>.
    monto = <fs_pr_tit>-precio.
    <valor> = monto.
ENDLOOP.

ADD 1 TO cont_prec.
IF cont_prec < 80.
    CONCATENATE 'WA_PLANTILLA1-COL' cont_prec INTO nombre.
ELSE.
    CONCATENATE 'WA_PLANTILLA2-COL' cont_prec INTO nombre.

```

```

ENDIF.
ASSIGN (nombre) TO <valor>.
<valor> = '/'.
ADD 1 TO cont_prec.
IF cont_prec < 80.
    CONCATENATE 'WA_PLANTILLA1-COL' cont_prec INTO nombre.
ELSE.
    CONCATENATE 'WA_PLANTILLA2-COL' cont_prec INTO nombre.
ENDIF.
ASSIGN (nombre) TO <valor>.
<valor> = '1'.
ADD 1 TO cont_prec.
IF cont_prec < 80.
    CONCATENATE 'WA_PLANTILLA1-COL' cont_prec INTO nombre.
ELSE.
    CONCATENATE 'WA_PLANTILLA2-COL' cont_prec INTO nombre.
ENDIF.
ASSIGN (nombre) TO <valor>.
<valor> = '/'.
ENDIF.

**OBTENER tabla de Valores y Restricciones. Config restricciones del Pro
ucto terminado
*   t_plantillas1  ztpp_valor1          codigo = codigo. "CDB7
*   t_plantillas2  ztpp_valor2          codigo = codigo. "CDB7

CLEAR lwa_config.
LOOP AT t_config INTO lwa_config.
ENDLOOP.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = '/'.
lwa_config-subtit = '/'.
lwa_config-caract = '/'.
lwa_config-subcar = '/'.
APPEND lwa_config TO t_config.
CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
LOOP AT lt_config_aux.
    ADD 1 TO lwa_config-columna.
    lwa_config-titulo = lt_config_aux-titulo.
    APPEND lwa_config TO t_config.
ENDLOOP.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = '/'.

```

```

lwa_config-subtit = '/'.
lwa_config-caract = '/'.
lwa_config-subcar = '/'.
APPEND lwa_config TO t_config.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = 'TEXTO'.
CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
APPEND lwa_config TO t_config.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = '/'.
lwa_config-subtit = '/'.
lwa_config-caract = '/'.
lwa_config-subcar = '/'.
APPEND lwa_config TO t_config.
ADD 1 TO lwa_config-columna.
lwa_config-titulo = 'ESTANDAR'.
CLEAR: lwa_config-subtit, lwa_config-caract, lwa_config-subcar.
APPEND lwa_config TO t_config.

```

```

**DETERMINAR PLANTILLA STANDARD**

```

```

*   te_plantillas2 INTO wea_plantilla2 WITH KEY standard = 'X'.
*   = 0.
**       IF wa_config-texto IS NOT INITIAL.
*   dmp200_m_text_s = wa_plantilla2-texto.
*   dmp200_m_text   = wa_plantilla2-texto.
**       ENDIF.
*   t_plantillas1 INTO wta_plantilla1 WITH KEY fila = wa_plantilla2-
fila.
*   ENDIF.

```

```

"LA TABLA DE FLAGS SE MANTIENE.

```

```

DO 140 TIMES.
  ADD 1 TO cont.
  IF cont < 80.
    CONCATENATE 'wa_flag1-col' cont INTO nombre.
    ASSIGN (nombre) TO <fs_val>.
    <fs_val> = 'X'.

  ELSE.
    CONCATENATE 'wa_flag2-col' cont INTO nombre.

```

```

        ASSIGN (nombre) TO <fs_val>.
        <fs_val> = 'X'.

    ENDIF.
ENDDO.

PERFORM formatear_plantilla TABLES gt_cotiza_outtab.

ENDFORM.
FORM mostrar_cotizacion .
*   DATA: wa_output TYPE ty_plantilla.
DATA: lt_config TYPE STANDARD TABLE OF ztpp_config.
DATA: wa_config TYPE ztpp_config.
DATA: flag_editable(1) TYPE c.
DATA: lwa_cotiza_outtab TYPE gty_cotiza_outtab.
REFRESH:  gt_cotiza_outtab_alv,gt_cotiza_outtab.
DATA: lw_longitud TYPE ztpp_longitud.

REFRESH gt_cotiza_outtab.
*   READ TABLE gt_plantillas INTO wa_output INDEX e_row-index.
IF gwa_plant IS NOT INITIAL.
    gv_codigo = gwa_plant-codigo.
    SELECT * FROM ztpp_config INTO TABLE lt_config WHERE
    codigo = gwa_plant-codigo AND
    secuen_config = gwa_plant-secuen_config.

    IF sy-subrc = 0.
*dmp200_numcot
        gv_cod_ref2      = gwa_plant-codigo.
        gv_secuen_ref2   = gwa_plant-secuen_config.
        dmp200_tip_vta   = gwa_plant-tip_vta.
        dmp200_matnr     = gwa_plant-matnr.
        dmp200_mod_bus   = gwa_plant-mod_bus.
        dmp200_tip_car   = gwa_plant-tip_car.
        dmp200_vers      = gwa_plant-vers.

```

```

dmp200_ano_mod = gwa_plant-ano.
CLEAR lw_longitud .

SELECT SINGLE * INTO lw_longitud FROM ztpp_longitud
WHERE matnr = dmp200_matnr .

```

```

dmp200_long          = lw_longitud-longitud.
dmp200_ancho        = lw_longitud-ancho.
dmp200_alto         = lw_longitud-alto.
dmp200_dee          = lw_longitud-dee.
dmp200_vol_del      = lw_longitud-vol_del.
dmp200_vol_post     = lw_longitud-vol_post.
dmp200_norma_r66    = lw_longitud-norma_r66.

```

```

*dmp200_cant        = wa
*dmp200_kunnr
*dmp200_kunnr2
*dmp200_land1
*dmp200_ano_mod    = wa_output-
*dmp200_ano_fab

```

```

LOOP AT lt_config INTO wa_config.
  IF wa_config-texto IS NOT INITIAL.
    dmp200_m_text_s = wa_config-texto.
  ENDIF.
  CLEAR lwa_cotiza_outtab.
  MOVE-CORRESPONDING wa_config TO lwa_cotiza_outtab.
  lwa_cotiza_outtab-st_fila = ''.
  lwa_cotiza_outtab-icon = '@5B@'.
  APPEND lwa_cotiza_outtab TO gt_cotiza_outtab.
*
*      PERFORM valor_editable USING wa_config-titulo wa_config-
subtitulo wa_config-caracterica wa_config-subcaracteristica
*
*      CHANGING flag_editable.
*
*      @5B@
*
*      @04@
*
*      @11@
ENDLOOP.

```



```

*      CALL SCREEN 200.
      ENDIF.

*      PERFORM mostrar_detalle.
      ENDIF.

ENDFORM.

*&-----*
*&      Form  DESCUENTO
*&-----*
*      text
*-----*
*  --> p1      text
* <-- p2      text
*-----*

FORM descuento .
      CALL SELECTION-SCREEN 008 STARTING AT 20 5.

ENDFORM.

*-----*
*      For  CAMPOS_EDITABLES
*-----*

FORM campos_editables .
      DATA: wa_ucomm_aux TYPE c.

      LOOP AT SCREEN.
          IF screen-name = 'DMP200_NUMCOT'

              .
*      IF ge_ucomm = 'C' OR ge_ucomm = 'F' .
screen-input = 0.
*      ELSEIF ge_ucomm = 'M' OR ge_ucomm = 'V' ..
screen-input = 1.
*      ENDIF.
      ENDIF.

*      DMP200_PRODUCTO

```

```

IF screen-name = 'DMP200_KUNNR' OR screen-name = 'DMP200_KUNNR2'
  OR screen-name = 'DMP200_L_KUNNR' OR screen-
name = 'DMP200_L_KUNNR2' OR screen-name = 'DMP200_L_VENDEDOR'
  OR screen-name = 'DMP200_ANO_FAB' OR screen-
name = 'DMP200_CANT' OR screen-name = 'DMP200_TIP_COT' OR screen-
name = 'DMP200_INCOTERM'
  OR screen-name = 'DMP200_F_V_COT_D' OR screen-
name = 'DMP200_F_V_COT_H' OR screen-name = 'DMP200_PLAZO_ENT' OR screen-
name = 'DMP200_TIP_PLACA'
  OR screen-name = 'DMP200_ANO_FAB_CARR' OR screen-
name = 'DMP200_OBSERV_COTIZA'.
  IF ge_ucomm = 'C' OR ge_ucomm = 'M' .
    screen-input = 1.
  ELSEIF ge_ucomm = 'V' .
    screen-input = 0.
  ENDIF.
ENDIF.

IF screen-name = 'DMP200_PRODUCTO'.
  IF ge_ucomm = 'N' OR ge_ucomm = 'F'.
    screen-input = 1.
  ELSE.
    screen-input = 0.
  ENDIF.
ENDIF.

IF screen-name = 'DMP200_ZTERM'.
  IF ge_ucomm = 'N' OR ge_ucomm = 'M' OR ge_ucomm = 'F'.
    screen-input = 1.
  ELSE.
    screen-input = 0.
  ENDIF.
ENDIF.

IF screen-name = 'DMP200_INCOTERM' OR screen-
name = 'L_INCOTERM' OR screen-name = 'DMP200_L_INCOTERM'.
  IF dmp200_tip_cot <> '3'.
    CLEAR dmp200_incoterm.
*      CLEAR dmp200_l_incoterm.      ENDIF.

ENDIF.

IF screen-name = 'DMP200_VENDEDOR'.

```

```

ENDIF.

IF screen-group1 = 'DTP'.

    IF sy-subrc <> 0.
        screen-invisible = 1.
    ENDIF.
endif.

MODIFY SCREEN.
ENDLOOP.
IF dmp200_kunnr = '99999'.
    dmp200_l_kunnr = dmp300_c_nombre.
ENDIF.
IF c_usd = 'X'.
    l_m1 = 'USD'.
    l_m2 = 'USD'.
    l_m3 = 'USD'.
    l_m4 = 'USD'.
    l_m5 = 'USD'.
    l_m6 = 'USD'.
    l_m7 = 'USD'.
ENDIF.
IF c_pen = 'X'.
    l_m1 = 'PEN'.
    l_m2 = 'PEN'.
    l_m3 = 'PEN'.
    l_m4 = 'PEN'.
    l_m5 = 'PEN'.
    l_m6 = 'PEN'.
    l_m7 = 'PEN'.
ENDIF.
IF ge_ucomm = 'N' OR ge_ucomm = 'F'.
    IF dmp200_f_v_cot_d IS INITIAL AND dmp200_f_v_cot_d IS INITIAL.
        dmp200_f_v_cot_d = sy-datum.
        dmp200_f_v_cot_h = sy-datum + gv_vig_cot.
    ENDIF.
    IF dmp200_plazo_ent IS INITIAL.
        dmp200_plazo_ent = gv_plazo_ent.
    ENDIF.
    IF dmp200_ano_fab_carr IS INITIAL.

```

```

        dmp200_ano_fab_carr = sy-datum(4).
    ENDIF.
ENDIF.

ENDFORM.

*-----*
*      OBTENER_COTIZACION
*-----*

FORM obtener_cotizacion .
*   PERFORM obtener_nro_cotizacion USING dmp200_numcot.
DATA: dmpv_message(80) TYPE c.
*   DATA: dmpv_message(80) TYPE c.
DATA: dmpv_version2 TYPE c.
*   SELECT SINGLE version2 FROM ztpp_cotiza_cab INTO lv_version2 WHERE nu
m_cot = dmp200_numcot.

SELECT SINGLE   cod_ref kunnr kunnr2 land1 zterm vendedor tip_vta
matnr mod_bus tip_car cantidad
ano_mod ano_fab tip_placa longitud alto vol_del ancho dee vol_post p
v_netto flete igv dcto cost_serv cost_man_ob pv_final
marg_ganancha_r marg_gananc_r marg_ganancha marg_gananc ruc dni nombre
_cli contac correo telefono
direccion prec_chasis prec_carro pv_netto_r igv_r incoterm moneda t_c
amb kunnr2_l vendedor_l aceptado tp_cotiza f_v_cot_d f_v_cot_h
plazo_ent secuen_ref secuen_ref2 confirmado aprobado rechazado zalias
cost_seguridad observ_cotiza ano_fab_carr version2

INTO

(dmp200_producto, dmp200_kunnr, dmp200_kunnr2, dmp200_land1, dmp2
00_zterm, dmp200_vendedor, dmp200_tip_vta, dmp200_matnr, dmp200_mod_bu
s, dmp200_tip_car,
dmp200_cant, dmp200_ano_mod, dmp200_ano_fab, gv_tip_placa, dmp200
_long, dmp200_alto, dmp200_vol_del, dmp200_ancho, dmp200_dee, dmp200
_vol_post,
dmp200_prec_netto_f, dmp200_flete_f, dmp200_igv_f, dmp200_dcto_f, g
v_cost_serv, gv_cost_man_ob, dmp200_pv_total, dmp200_margen, p_marg,
dmp200_particip_chasis_f, dmp200_particip_carroc_f,
dmp300_c_ruc, dmp300_c_dni, dmp300_c_nombre, dmp300_c_contacto, d

```

```

mp300_c_correo, dmp300_c_telefono, dmp300_c_direccion, dmp200_prec_cha
sis,
    dmp200_prec_carroc, dmp200_prec_netto, dmp200_igv,dmp200_incoterm, l
_m1, p_t_camb,dmp200_l_kunnr2,dmp200_l_vendedor,dmp200_aceptado,dmp200_ti
p_cot,dmp200_f_v_cot_d,dmp200_f_v_cot_h,dmp200_plazo_ent,dmp200_m_text_s,
dmp200_m_text,
    dmp200_confirmado, dmp200_aprobado, dmp200_rechazado,dmp200_moduni, g
v_cost_seguridad, dmp200_observ_cotiza, dmp200_ano_fab_carr, lv_version2)

    FROM ztpp_cotiza_cab
    WHERE num_cot = dmp200_numcot.

dmp200_tip_placa = gv_tip_placa.

IF l_m1 = 'USD'.
    c_usd = 'X'.
ELSEIF l_m1 = 'PEN'.
    c_pen = 'X'.
ENDIF.

l_m2 = l_m1.
l_m3 = l_m1.
l_m4 = l_m1.
l_m5 = l_m1.
l_m6 = l_m1.

p_descu = dmp200_dcto_f.
p_flete = dmp200_flete_f.

IF sy-subrc NE 0 AND ( ge_ucomm = 'V' OR ge_ucomm = 'M' ).
    CONCATENATE 'No Existe Cotizacion' dmp200_numcot INTO lv_message.
    CLEAR mp200_estado_cotiza.
    MESSAGE lv_message TYPE 'S' DISPLAY TYPE 'E'.
ELSE.
    IF lv_version2 EQ 'X'. "ADD DB7
        IF dmp200_aceptado = 'X'.
            mp200_estado_cotiza = 'Liberado'.
        ELSE.
            mp200_estado_cotiza = 'Bloqueado'.
        ENDIF.

        IF dmp200_confirmado = 'X'.

```

```

        mp200_estado_cotiza = 'Confirmado por Cliente'.
    ENDIF.

    IF dmp200_aprobado = 'X'.
        mp200_estado_cotiza = 'Aprobado por Comercial'.
    ENDIF.

    IF dmp200_rechazado = 'X'.
        mp200_estado_cotiza = 'Rechazado'.
    ENDIF.

    dmp300_c_kunnr = dmp200_kunnr.
*   CLEAR: DMP300_C_KUNNR.
ELSE.
    PERFORM cotizador USING dmp200_numcot.
    PERFORM limpiar_campos.
    PERFORM limpiar_montos .
    CONCATENATE 'Tratar Cotizacion' dmp200_numcot 'Desde ZCOTIZADORBUS'
INTO lv_message SEPARATED BY space.
    CLEAR mp200_estado_cotiza.
    MESSAGE lv_message TYPE 'S' DISPLAY TYPE 'E'.
    ENDIF.          "ADD DB7
ENDIF.

*   REFRESH gt_cotiza_outtab.
*DMP200_PRODUCTO
    IF lv_version2 EQ 'X'.
        DATA: lwa_cotiza_det1 TYPE  ztpc_cotiza_det1.
        DATA: lwa_cotiza_det2 TYPE  ztpc_cotiza_det2.
        DATA: lwa_cotiza_st1  TYPE  ztpc_cotiza_st1.
        DATA: lwa_cotiza_st2  TYPE  ztpc_cotiza_st2.
        DATA: lwa_cotiza_conf TYPE STANDARD TABLE OF ztpc_cotiza_conf WITH HE
ADER LINE.

        SELECT SINGLE * FROM ztpc_cotiza_det1 INTO  lwa_cotiza_det1 WHERE num
_cot = dmp200_numcot.
        SELECT SINGLE * FROM ztpc_cotiza_det2 INTO  lwa_cotiza_det2 WHERE num
_cot = dmp200_numcot.
        SELECT SINGLE * FROM ztpc_cotiza_st1 INTO  lwa_cotiza_st1 WHERE num_c
ot = dmp200_numcot.
        SELECT SINGLE * FROM ztpc_cotiza_st2 INTO  lwa_cotiza_st2 WHERE num_c
ot = dmp200_numcot.

```

```

SELECT * FROM ztpp_cotiza_conf INTO TABLE lwa_cotiza_conf WHERE num_c
ot = dmp200_numcot.
SELECT * FROM ztpp_restr1 INTO TABLE t_restr1 WHERE codigo = dmp200_pr
oducto.
SELECT * FROM ztpp_restr2 INTO TABLE t_restr2 WHERE codigo = dmp200_pr
oducto.
SELECT * FROM ztpp_configura2 INTO CORRESPONDING FIELDS OF TABLE t_c
onfigu_2 WHERE codigo = dmp200_producto.
SELECT * FROM ztpp_valor_carac INTO TABLE t_vc WHERE codigo = dmp20
0_producto AND valor NE space.
SELECT * FROM ztpp_valor_carac INTO TABLE t_vc2 WHERE codigo = dmp2
00_producto AND valor EQ space.

REFRESH t_config.
CLEAR: wa_plantilla1,wa_plantilla2,wa_flag1,wa_flag2.
LOOP AT lwa_cotiza_conf .
    CLEAR t_config.
    MOVE-CORRESPONDING lwa_cotiza_conf TO t_config.
*      t_config-num_cot = dmp200_numcot.
    APPEND t_config.
ENDLOOP.
MOVE-CORRESPONDING lwa_cotiza_det1 TO wa_plantilla1.
MOVE-CORRESPONDING lwa_cotiza_det2 TO wa_plantilla2.
MOVE-CORRESPONDING lwa_cotiza_st1 TO wa_flag1.
MOVE-CORRESPONDING lwa_cotiza_st2 TO wa_flag2.
ENDIF.
PERFORM formatear_plantilla TABLES gt_cotiza_outtab.

*      gt_cotiza_outtab zpp_cotiza_det
*      num_cot = dmp200_numcot.

CLEAR: c_usd, c_pen.

IF l_m1 = 'USD'.
    c_usd = 'X'.
ELSEIF l_m1 = 'PEN'.
    c_pen = 'X'.
ENDIF.

```

ENDFORM.

```
*&-----*
*&      Form  OBTENER_PLANT_ST
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
```

FORM obtener_plant_st .

```
TYPES: BEGIN OF ty_precios,
        codigo      TYPE ze_pp_cod_clave,
        secuen_config TYPE ze_pp_secuen_config,
        ord_tit      TYPE ze_pp_ord_tit,
        prec_tit     TYPE ze_pp_prec_tit,
END OF ty_precios.
```

```
DATA: lt_precios TYPE STANDARD TABLE OF ty_precios WITH HEADER LINE.
```

```
DATA: lwa_constant TYPE ztpp_constant.
```

```
FIELD-SYMBOLS: <fe_plant_st> TYPE LINE OF gt_plantillas_st.
```

```
* DELETE FROM ztpp_claves WHERE CODIGO = '2000000007'.
```

```
SELECT a~mandt a~codigo a~tip_vta a~ano a~mod_bus a~tip_car a~vers a~mar_cha
```

```
        a~mod_cha a~cobust a~motor a~mod_motor a~form_rod a~matnr
```

```
INTO TABLE gt_plantillas_st
```

```
FROM ztpp_claves AS a . "AUT_PRE
```

```
* SELECT a~mandt a~codigo a~tip_vta a~ano a~mod_bus a~tip_car a~vers a~mar_cha
```

```
*      a~mod_cha a~cobust a~motor a~mod_motor a~form_rod a~matnr a~ocult
```

```
*      INTO TABLE gt_plantillas_st
```

```
*      FROM ztpp_claves AS a .
```

```
*      a~tip_vta = p_tvta AND
```

```
*      a~ano = p_ano AND
```

```
*      a~mod_bus = p_mbus AND
```

```
*      a~tip_car = p_tcar AND
```

```
*      a~vers = p_vers AND
```

```
*      a~mar_cha = p_macha AND
```

```
*      a~mod_cha = p_mocha AND
```

```
*      a~cobust = p_comb AND
```

```
*      a~motor = p_mot AND
```

```
*      a~mod_motor = p_m_mot.
```



```

IF sy-subrc = 0.
    SORT gt_plantillas_st BY codigo secuen_config.
    DELETE ADJACENT DUPLICATES FROM gt_plantillas_st COMPARING codigo sec
uen_config.
    LOOP AT gt_plantillas_st ASSIGNING <fs_plant_st>.
        REFRESH lt_precios.
*         CLEAR: <fs_plant_st>-cost_serv.
*         CLEAR: <fs_plant_st>-cost_man_ob.
*         CLEAR: <fs_plant_st>-pv_netto.

SELECT codigo secuen_config ord_tit prec_tit
FROM ztpp_config
INTO TABLE lt_precios
WHERE codigo = <fs_plant_st>-codigo AND
        secuen_config = <fs_plant_st>-secuen_config.
IF sy-subrc = 0.
*     LOOP AT lt_precios.
*         ADD lt_precios-prec_tit TO <fs_plant_st>-pv_netto.
*     ENDLOOP.

ENDIF.
CLEAR lwa_constant.
SELECT SINGLE * FROM ztpp_constant INTO lwa_constant WHERE modulo
= 'PP' AND
                                aplicacion = 'CO
TIZACION' AND
                                programa = 'COST
O_SERVICIO' AND
                                campo = 'COST
_SERVICE' AND
                                valor1 = <fs_p
lant_st>-mod_bus.
IF sy-subrc = 0.
*     <fs_plant_st>-cost_serv = lwa_constant-valor2.
ENDIF.

CLEAR lwa_constant.
SELECT SINGLE * FROM ztpp_constant INTO lwa_constant WHERE modulo
= 'PP'
                                AND aplicacion =
' COTIZACION'

```

```

AND programa =
' COSTO_MANO_OBRA'
AND campo =
' COST_MOD_MOI'
AND valor1 =
<fs_plant_st>-mod_bus
AND valor2 =
<fs_plant_st>-tip_car.
IF sy-subrc = 0.
* <fs_plant_st>-cost_man_ob = lwa_constant-valor3.
ENDIF.

ENDLOOP.

* LOOP AT gt_plantillas_st INTO gwa_plant WHERE standard = 'X'.
* EXIT.
* ENDLOOP.

* IF gwa_plant IS NOT INITIAL.

* ELSE.
* READ TABLE gt_plantillas INTO gwa_plant INDEX 1.
* ENDIF.

ENDIF.

ENDFORM.

*&-----*
*& Form CREAR_COTIZACION
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text
*-----*
*&-----*
*& Form VISUALIZAR_COTIZACION
*&-----*

```

```

*          text
*-----*
* --> p1          text
* <-- p2          text
*-----*
FORM visualizar_cotizacion .
* CALL SELECTION-SCREEN 009 STARTING AT 20 5.
  IF p_n_cot IS NOT INITIAL.
    ENDIF.
    ge_ucomm = 'B'.
    CLEAR: p_clie, p_vend, p_ano, l_clie, l_vend.

ENDFORM.
*&-----*
*&          Form  LIMPIAR_CAMPOS
*&-----*
*          text
*-----*
* --> p1          text
* <-- p2          text
*-----*
FORM limpiar_campos .
* IF ge_ucomm EQ 'N' OR ge_ucomm EQ 'F'.
*   CLEAR: dmp200_numcot .
* ENDIF.
  CLEAR: p_clie, p_vend, p_ano, l_clie, l_vend, dmp200_tip_vta , dmp200_ma
tnr, p_descu
          , dmp200_mod_bus, dmp200_tip_car, dmp200_vers, dmp200_cant, dmp200_ku
nnr, dmp200_producto
          , dmp200_l_kunnr, dmp200_kunnr2, dmp200_l_kunnr2, dmp200_vendedor, dm
p200_l_vendedor, dmp200_land1
          , dmp200_ano_mod, dmp200_ano_fab, dmp200_observ_cotiza, dmp200_tip_p
laca, dmp200_moneda, dmp200_igv, dmp200_manoobra
          , dmp200_tmp_entrega, dmp200_precio, dmp200_costo_total, dmp200_dcto
, dmp200_pv_final, dmp200_pv_igv
          , dmp200_flete, dmp200_tp_camb, dmp200_precio_aux, dmp200_costo_tota
l_aux, dmp200_dcto_aux, dmp200_margen_aux
          , dmp200_pv_final_aux, dmp200_pv_igv_aux, dmp200_flete_aux, dmp200_t
p_camb_aux, dmp200_igv_aux
          , dmp200_long, dmp200_ancho, dmp200_alto, dmp200_dee, dmp200_vol_del,
dmp200_vol_post, dmp200_norma_r66
          , dmp200_chasis, dmp200_margen, dmp200_tip_placa, dmp200_cost_chasis

```

```
,dmp200_tip_placa,dmp200_prec_neto_f,
      dmp200_flete_f,dmp200_igv_f,dmp200_prec_bruto_f,dmp200_prec_brut
o_f,dmp200_prec_venta_f
      ,dmp200_pv_total,gv_tip_placa,dmp200_cost_chasis2,p_marg,dmp300_
c_ruc,dmp300_c_dni,
      dmp300_c_kunnr,dmp300_c_nombre,dmp300_c_contacto,dmp300_c_correo
,dmp300_c_telefono,dmp300_c_direccion,dmp200_zterm,dmp200_l_zterm
      ,dmp200_m_text_s,dmp200_m_text,dmp200_macha,dmp200_mocha,dmp200_
comb,dmp200_mot ,dmp200_m_mot,dmp200_f_rod,
      dmp200_l_land1, dmp200_f_v_cot_d, dmp200_f_v_cot_h,dmp200_plazo_
ent,dmp200_numcot_ref,mp200_estado_cotiza,dmp200_moduni
      ,dmp200_ano_fab_carr.
```

```
REFRESH: gt_cotiza_outtab,gt_cotiza_outtab_alv.
```

```
ENDFORM.
```

```
*&-----*
*&      Form  DATOS_CLIENTE
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
```

```
FORM datos_cliente .
```

```
CALL SCREEN 300 STARTING AT 45 5
```

```
ENDING AT 100 15.
```

```
ENDFORM.
```

```
*&-----*
*&      Form  CALCULAR_PARTICIPACION_MARGEN
*&-----*
*      text
*-----*
*      -->P_DMP200_CHASIS  text
*      -->P_DMP200_COSTO_TOTAL  text
*      -->P_DMP200_CHASIS_MARG  text
*      -->P_DMP200_COSTO_TOTAL_MARG  text
*      <--P_DMP200_PARTICIP_CHASIS  text
*      <--P_DMP200_PARTICIP_CARROC  text
*-----*
```

```
FORM calcular_participacion_margen USING      p_dmp200_chasis
                                              p_dmp200_costo_total
                                              p_dmp200_chasis_marg
```

```

p_dmp200_costo_total_marg
CHANGING p_dmp200_particip_chasis
p_dmp200_particip_carroc.

DATA: p1 TYPE ze_pp_precio.
DATA: p2 TYPE ze_pp_precio.
DATA: p3 TYPE ze_pp_precio.

p1 = p_dmp200_chasis_marg - p_dmp200_chasis.
p2 = p_dmp200_costo_total_marg - p_dmp200_costo_total.
p3 = p1 + p2.
IF p3 IS NOT INITIAL.
    p_dmp200_particip_chasis = p1 * 100 / p3.
ENDIF.
p_dmp200_particip_carroc = 100 - p_dmp200_particip_chasis.

ENDFORM.

*&-----*
*&      Form  CALCULAR_MARGEN
*&-----*
*      text
*-----*
*      -->P_DMP200_PARTICIP_CHASIS  text
*      -->P_DMP200_PARTICIP_CARROC  text
*      <--P_DMP200_PARTICIP_CHASIS_F  text
*      <--P_DMP200_PARTICIP_CARROC_F  text
*-----*

FORM calcular_margen USING    p_dmp200_particip_chasis
                             p_dmp200_particip_carroc
                             p_dmp200_costo_total
                             p_dmp200_chasis
                             CHANGING p_dmp200_particip_chasis_f
                             p_dmp200_particip_carroc_f.

* *dmp200_prec_netto_f
*dmp200_prec_bruto_f
*dmp200_prec_venta_f
*dmp200_igv_f
*dmp200_dcto_f
*dmp200_flete_f

```

```

DATA: marg_total TYPE ze_pp_precio.
DATA: s1 TYPE ze_pp_precio.
DATA: s2 TYPE ze_pp_precio.
marg_total = dmp200_prec_venta_f - dmp200_igv_f - dmp200_flete_f - p_dm
p200_costo_total - p_dmp200_chasis.
* marg_total = dmp200_prec_venta - dmp200_igv - dmp200_flete - p_dmp200_
costo_total - p_dmp200_chasis.
p_dmp200_particip_chasis_f = ( marg_total * p_dmp200_particip_chasis )
/ 100.
p_dmp200_particip_carroc_f = marg_total - p_dmp200_particip_chasis_f.
s1 = ( p_dmp200_chasis + p_dmp200_particip_chasis_f ).
s2 = ( p_dmp200_costo_total + p_dmp200_particip_carroc_f ).
IF s1 IS NOT INITIAL.
p_dmp200_particip_chasis_f = 100 - p_dmp200_chasis / s1 * 100.
ENDIF.
IF s2 IS NOT INITIAL.
p_dmp200_particip_carroc_f = 100 - p_dmp200_costo_total / s2 * 100.
ENDIF.
ENDFORM.

*&-----*
*& Form SEL_TIP_COTIZA
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text
*-----*

FORM sel_tip_cotiza .
*LOOP AT SCREEN.
* IF SCREEN-name = 'L_FLETE'.
* SCREEN-name = 'DFD'.
* ENDIF.
* MODIFY SCREEN.
*ENDLOOP.

IF dmp200_tip_cot = '3'.
l_flete = 'Gastos Exp.'.
* LOOP AT SCREEN.
* IF SCREEN-name = 'DMP200_INCOTERM'.

```

```

**      SCREEN-INPUT = '1'.
**      SCREEN-OUTPUT = '0'.
**      SCREEN-invisible = '1'.
*      SCREEN-active = '0'.
*      MODIFY SCREEN.
*      ENDIF.
*      ENDLOOP.
ELSE.
    l_flete = 'Flete'.
ENDIF.

ENDFORM.

FORM obtener_cabecera .
    CLEAR: class2, feature_info, r_fieldcat2, gt_data.
    REFRESH: feature_info, gt_fieldcat2, gt_data.
    DATA: cont TYPE i,
           scont TYPE string.

    class2 = 'Z_SVEH'.

    CALL FUNCTION 'CLAS_READ_FEATURE_INFO'
        EXPORTING
            class          = class2
            classtype     = '300'
*       LANGUAGE        = SY-LANGU
*       KEY_DATE       = SY-DATUM
    TABLES
        feature_info     = feature_info[]
    EXCEPTIONS
        class_not_exist = 1
        OTHERS          = 2.
    IF sy-subrc = 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*       WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
    LOOP AT feature_info.

```

```

    ADD 1 TO cont.
    scont = cont.
    gt_data-descr = feature_info-smbez.
    CONCATENATE 'COL' scont INTO r_fieldcat2-fieldname.
    r_fieldcat2-reptext = feature_info-smbez.
    r_fieldcat2-outputlen = 50.
    APPEND r_fieldcat2 TO gt_fieldcat2.
    CLEAR r_fieldcat2.
    APPEND gt_data.
ENDLOOP.

ENDIF.

CALL METHOD cl_alv_table_create=>create_dynamic_table
EXPORTING
    it_fieldcatalog = gt_fieldcat2[]
IMPORTING
    ep_table        = t_newtable.

CALL METHOD cl_alv_table_create=>create_dynamic_table
EXPORTING
    it_fieldcatalog = gt_fieldcat2[]
IMPORTING
    ep_table        = t_newtable2.

ASSIGN t_newtable->* TO <t_dyntable>.
ASSIGN t_newtable2->* TO <t_dyntable_mach>.
* Create dynamic work area and assign to FS
CREATE DATA t_newline TYPE LINE OF <t_dyntable>.
PERFORM obtener_data.

ENDFORM.

FORM obtener_data .
    DATA: BEGIN OF gt_atwrt OCCURS 0,
            atwrt TYPE atwrt,
            objek TYPE objnum,
            END OF gt_atwrt.
    DATA: v_atwrt TYPE atwrt.

    DATA: dmpv_matnr TYPE matnr,

```



```

lv_mstae TYPE mstae,
v_atflv TYPE ausp-atflv,
v_costo TYPE p DECIMALS 2.

```

```
FIELD-SYMBOLS: <feature_info> TYPE LINE OF feature_info.
```

```
READ TABLE feature_info ASSIGNING <feature_info> WITH KEY smbez = 'UBI
CACION TECNICA'.
```

```

SELECT atwrt objek INTO TABLE gt_atwrt FROM ausp
  WHERE atinn = <feature_info>-atinn.
ASSIGN t_newline->* TO <fs_dyntable>.
LOOP AT gt_atwrt.
  LOOP AT feature_info ASSIGNING <feature_info>.
    ASSIGN COMPONENT sy-tabix
      OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
  CLEAR v_atwrt.
  SELECT SINGLE atwrt INTO v_atwrt FROM ausp
  WHERE atinn = <feature_info>-atinn AND
    objek = gt_atwrt-objek.

```

```
<fs_fldval> = v_atwrt.
```

```

IF <feature_info>-smbez = 'COSTO MANO OBRA' OR <feature_info>-
smbez = 'COSTO SERVICIO' OR <feature_info>-
smbez = 'MARGEN INFERIOR CHASIS' OR
  <feature_info>-smbez = 'MARGEN SUPERIOR CHASIS' OR <feature_info>-
smbez = 'MARGEN CHASIS' OR <feature_info>-smbez = 'MARGEN CARROCERIA'
  OR <feature_info>-smbez = 'COSTO SEGURIDAD'...

```

```

SELECT SINGLE atflv INTO v_atflv FROM ausp
WHERE atinn = <feature_info>-atinn AND
  objek = gt_atwrt-objek.
IF sy-subrc = 0.
  v_costo = v_atflv.
  <fs_fldval> = v_costo.
  CONDENSE <fs_fldval> NO-GAPS.
ENDIF.
ENDIF.

```

```

        IF <feature_info>-smbez = 'UBICACION TECNICA'.
            lv_matnr = v_atwrt.
        ENDIF.
    ENDLOOP.

    SELECT SINGLE mstae INTO lv_mstae FROM mara
    WHERE matnr = lv_matnr.
    IF lv_mstae <> 'ZZ'.

        APPEND <fs_dyntable> TO <t_dyntable>.
    ENDIF.
ENDLOOP.
* <t_dyntable_mach> = <t_dyntable>.

ENDFORM.

FORM busca_mach2 USING v_tip_vta v_mod_bus v_tip_car v_mar_cha v_mod_cha
                    v_cobust v_motor v_mod_motor v_form_rod CHANGING er
ror matnr.
    DATA: dmpv_error TYPE c.
*   FIELD-
SYMBOLS: <t_dyntable>          TYPE STANDARD TABLE, " *          <fs_
dyntable>          TYPE any, " Field symbol to create work area
*
*
*          <t_dyntable_mach> TYPE STANDARD TABLE.

REFRESH <t_dyntable_mach>.

LOOP AT <t_dyntable> ASSIGNING <fs_dyntable>.
* read table <t_dyntable> index es_row_no-
row_id assigning <fs_dyntable>.
    CLEAR lv_error .
    ASSIGN COMPONENT 1 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   dmp300_catun IS NOT INITIAL.

```

```

IF <fs_fldval> <> v_tip_vta ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 2 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs_fldval> <> v_tip_car ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 4 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs_fldval> <> v_mod_bus ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 5 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs_fldval> <> v_mar_cha ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 6 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs_fldval> <> v_mod_cha ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

*   ASSIGN COMPONENT 5 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
**   dmp300_catun IS NOT INITIAL.
*   IF <fs_fldval> <> v_mod_cha ."dmp300_catun.
*     lv_error = 'X'.
*   ENDIF.

ASSIGN COMPONENT 11 OF STRUCTURE <fe_dyntable> TO <fe_fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs_fldval> <> v_cobust ."dmp300_catun.
  lv_error = 'X'.
ENDIF.

```

```

ASSIGN COMPONENT 7 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_motor ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 8 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_mod_motor ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 13 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_form_rod ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 14 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> matnr ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

IF lv_error IS INITIAL.
    EXIT.
ENDIF.

ENDLOOP.

IF lv_error = 'X'.
    error = 'X'.
    CLEAR matnr.
ELSE.
    error = ''.
    ASSIGN COMPONENT 14 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
    IF matnr <> <fe flval>.

```

```

        CLEAR matnr.
    ENDIF.
ENDIF.

ENDFORM.

FORM busca_mach USING v_tip_vta v_mod_bus v_tip_car v_mar_cha v_mod_cha
                    v_cobust v_motor v_mod_motor v_form_rod .

DATA: dmpv_error TYPE c.
REFRESH <t_dyntable_mach>.

LOOP AT <t_dyntable> ASSIGNING <fs_dyntable>.
* <t_dyntable> assigning <fe_dyntable>.
    CLEAR lv_error .
    ASSIGN COMPONENT 1 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
* IF dmp300_catun IS NOT INITIAL.
    IF <fs fldval> <> v_tip_vta ."dmp300_catun.
        lv_error = 'X'.
    ENDIF.

    ASSIGN COMPONENT 2 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
* dmp300_catun IS NOT INITIAL.
    IF <fs fldval> <> v_tip_car ."dmp300_catun.
        lv_error = 'X'.
    ENDIF.

    ASSIGN COMPONENT 4 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
* IF dmp300_catun IS NOT INITIAL.
    IF <fs fldval> <> v_mod_bus ."dmp300_catun.
        lv_error = 'X'.
    ENDIF.

    ASSIGN COMPONENT 5 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
* IF dmp300_catun IS NOT INITIAL.
    IF <fs fldval> <> v_mar_cha ."dmp300_catun.
        lv_error = 'X'.
    ENDIF.

```

```

ASSIGN COMPONENT 6 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_mod_cha ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 11 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_cobust ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 7 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_motor ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 8 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_mod_motor ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 13 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> v_form_rod ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

ASSIGN COMPONENT 14 OF STRUCTURE <fe_dyntable> TO <fe fldval>.
*   IF dmp300_catun IS NOT INITIAL.
IF <fs fldval> <> dmp200_matnr ."dmp300_catun.
    lv_error = 'X'.
ENDIF.

```

```

CLEAR gv_imagen.

IF lv_error IS INITIAL.
    ASSIGN COMPONENT 14 OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_matnr = <fe flval>.

    ASSIGN COMPONENT 15 OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_long = <fe flval>.

    ASSIGN COMPONENT 16 OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_alto = <fe flval>.

    ASSIGN COMPONENT 17
        OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_ancho = <fe flval>.

    ASSIGN COMPONENT 18
        OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_vol_del = <fe flval>.

    ASSIGN COMPONENT 19
        OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_vol_post = <fe flval>.

    ASSIGN COMPONENT 20
        OF STRUCTURE <fe_dyntable> TO <fe fldval>.

    dmp200_dee = <fe flval>.

    ASSIGN COMPONENT 9
        OF STRUCTURE <fe_dyntable> TO <fe fldval>.

```

```
dmp200_moduni = <fe_flval>.
```

```
ASSIGN COMPONENT 21  
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
gv_imagen = <fe_flval>.
```

```
ASSIGN COMPONENT 24  
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
gv_margen_chas_inf = <fe_flval>.
```

```
ASSIGN COMPONENT 25  
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
gv_margen_chas_sup = <fe_flval>.
```

```
IF ge_ucomm <> 'V'.  
  ASSIGN COMPONENT 22  
    OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
gv_cost_man_ob = <fe_flval>.
```

```
ASSIGN COMPONENT 23  
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
gv_cost_serv = <fe_flval>.
```

```
ASSIGN COMPONENT 26  
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
```

```
dmp200_margen = <fe_flval>.
```



```

ASSIGN COMPONENT 27
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.

p_marg = <fe_flval>.

ASSIGN COMPONENT 28
      OF STRUCTURE <fe_dyntable> TO <fe fldval>.
gv_cost_seguridad = <fe_flval>.

ENDIF.

EXIT.
ENDIF.

ENDLOOP.

ENDFORM.
*&-----*
*&      Form  VALIDAR_OBLIGATORIOS
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM validar_obligatorios .
  CLEAR gv_validacion.
  DATA:lv_message(80) TYPE c.
  DATA:dmpv_cli TYPE kunnr.
  DATA:dmpv_clie TYPE kunnr.
  DATA:dmpv_STCD1 type KNA1-STCD1.
  DATA:dmpv_STCDT type KNA1-STCDT.
  DATA:dmpv_FVENC type ZSD_LAVADO_ACT-FVENC.

  IF dmp200_tip_cot IS INITIAL.

```

```

SET CURSOR FIELD 'DMP200_TIP_COT'.
gv_validacion = 'X'.
lv_message = 'Ingreso Tipo de Cotizacion'.
ENDIF.

IF dmp200_producto IS INITIAL.
SET CURSOR FIELD 'DMP200_PRODUCTO'.
gv_validacion = 'X'.
lv_message = 'Ingreso Modelo de Cotizacion'.
ENDIF.

IF dmp200_kunnr IS INITIAL.
SET CURSOR FIELD 'DMP200_KUNNR' LINE 5.
gv_validacion = 'X'.
lv_message = 'Ingreso Cliente'.
ELSE.
lv_clie = dmp200_kunnr.

SELECT SINGLE kunnr FROM knal INTO lv_clie WHERE kunnr = lv_clie.
IF sy-subrc <> 0.
SET CURSOR FIELD 'DMP200_KUNNR' LINE 5.
gv_validacion = 'X'.
lv_message = 'El Cliente no Existe'.
ENDIF.
ENDIF.

IF dmp200_kunnr2 IS INITIAL.
SET CURSOR FIELD 'DMP200_KUNNR2'.
gv_validacion = 'X'.
lv_message = 'Ingreso Cliente Final'.
ELSE.
lv_clie = dmp200_kunnr2.

SELECT SINGLE kunnr FROM knal INTO lv_clie WHERE kunnr = lv_clie.
IF sy-subrc <> 0.
SET CURSOR FIELD 'DMP200_KUNNR' LINE 5.
gv_validacion = 'X'.
lv_message = 'El Cliente Final no Existe'.
ENDIF.
ENDIF.

*****ADD DB*****

SELECT SINGLE STCD1 STCDT into (lv_STCD1 ,lv_STCDT) FROM knal WHERE k

```

```

unnr = lv_clie.
  if sy-subrc = 0.
    IN = lv_STCDT.
    SELECT SINGLE FVENC FROM ZSD_LAVADO_ACT INTO lv_FVENC WHERE docid = lv_STCD1 and tipdoc = lv_STCDT.
    if sy-subrc = 0.
      if lv_FVENC > sy-datum.
        gv_validacion = 'X'.
        lv_message = 'Cliente Sospechoso por Lavado de Activos'.
      endif.
    endif.
  endif.
endif.

```

*****ADD DB*****

ENDIF.

```

IF dmp200_vendedor IS INITIAL.
  SET CURSOR FIELD 'DMP200_VENDEDOR'.
  gv_validacion = 'X'.
  lv_message = 'Ingreso Vendedor'.
ENDIF.

```

```

IF dmp200_incoterm IS INITIAL AND dmp200_tip_cot = '3'.
  SET CURSOR FIELD 'DMP200_INCOTERM'.
  gv_validacion = 'X'.
  lv_message = 'Ingreso Incoterm'.
ENDIF.

```

```

IF dmp200_cant IS INITIAL.
  SET CURSOR FIELD 'DMP200_CANT'.
  gv_validacion = 'X'.
  lv_message = 'Ingreso Cantidad'.
ENDIF.

```

```

IF dmp200_f_v_cot_d IS INITIAL.
  SET CURSOR FIELD 'DMP200_F_V_COT_D'.
  gv_validacion = 'X'.
  lv_message = 'Ingreso Fechas de Vigencia de Cotización'.
ENDIF.

```

```

IF dmp200_f_v_cot_h IS INITIAL.
  SET CURSOR FIELD 'DMP200_F_V_COT_H'.

```

```

gv_validacion = 'X'.
lv_message = 'Ingreso Fechas de Vigencia de Cotización'.
ENDIF.
IF dmp200_plazo_ent IS INITIAL.
SET CURSOR FIELD 'DMP200_PLAZO_ENT'.
gv_validacion = 'X'.
lv_message = 'Ingreso Plazo de Entrega'.
ENDIF.
IF dmp200_ano_fab IS INITIAL.
SET CURSOR FIELD 'DMP200_ANO_FAB'.
gv_validacion = 'X'.
lv_message = 'Ingreso Año de Fabricación'.
ENDIF.
IF dmp200_ano_fab_carr IS INITIAL.
SET CURSOR FIELD 'DMP200_ANO_FAB_CARR'.
gv_validacion = 'X'.
lv_message = 'Ingreso Año de Fabricación de carrocería'.
ENDIF.
IF dmp300_c_dni IS INITIAL.
SET CURSOR FIELD 'DMP300_C_DNI'.
gv_validacion = 'X'.
lv_message = 'Ingreso NIF del Solicitante'.
ENDIF.
IF dmp200_tip_placa IS INITIAL.
SET CURSOR FIELD 'DMP200_TIP_PLACA'.
gv_validacion = 'X'.
lv_message = 'Ingreso Tipo de Placa'.
ENDIF.
IF gv_validacion = 'X'.
MESSAGE lv_message TYPE 'S' DISPLAY TYPE 'E'.
ENDIF.
ENDFORM.
*&-----*
*&      Form  CAMPOS_EDITABLES_300
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM campos_editables_300 .
LOOP AT SCREEN.

```

```

IF screen-name = 'DMP300_C_NOMBRE'.
  IF dmp200_kunnr = '99999'.
    A = 1.
    MODIFY SCREEN.
  ENDIF.
ENDIF.

IF ge_ucomm = 'V'.
  IF screen-name = 'DMP300_C_NOMBRE' OR screen-name = 'DMP300_C_DNI'
    OR screen-name = 'DMP300_C_CONTACTO'OR screen-
name = 'DMP300_C_RUC'OR
    screen-name = 'DMP300_C_CORREO' OR screen-
name = 'DMP300_C_TELEFONO' OR
    screen-name = 'DMP300_C_DIRECCION'.

    screen-input = 0.
    MODIFY SCREEN.

  ENDIF.
ELSE.
  IF screen-name = 'DMP300_C_NOMBRE' OR screen-
name = 'DMP300_C_DNI' OR
    screen-name = 'DMP300_C_CONTACTO'OR screen-name = 'DMP300_C_RUC'OR
    screen-name = 'DMP300_C_CORREO' OR screen-
name = 'DMP300_C_TELEFONO' OR
    screen-name = 'DMP300_C_DIRECCION'.

    screen-input = 1.
    MODIFY SCREEN.

  ENDIF.
ENDIF.

ENDLOOP.

ENDFORM.

FORM limpiar_montos .
  CLEAR: dmp200_cost_chasis ,
        dmp200_cost_chasis2,
        dmp200_cost_carroc ,
        dmp200_prec_chasis ,

```

```

        dmp200_prec_carroc ,
        dmp200_prec_netto ,
        dmp200_prec_bruto ,
        dmp200_igv1      ,

        dmp200_prec_netto_f ,
        dmp200_prec_bruto_f,
        dmp200_prec_venta_f,
        dmp200_igv_f      ,
        dmp200_dcto_f     ,
        dmp200_flete_f    ,
        dmp200_pv_total,
        dmp200_igv,
gv_cost_man_ob,
gv_cost_serv,
p_flete.

```

ENDFORM.

FORM imprimir_cotizacion USING imprime.

```

DATA: dmpv_fmname TYPE rs38l_fnam,
      gs_ctrlop   TYPE ssfctrlop,
      gs_outopt   TYPE ssfcompop.

```

```

DATA: lt_detalle_cot TYPE STANDARD TABLE OF zpp_cotiza_det .

```

```

DATA: lw_detalle_cot TYPE zpp_cotiza_det.

```

```

DATA: lw_clave TYPE ztpp_claves.

```

```

DATA: dmpv_const_cotiza TYPE ztpp_constant .

```

* *DMP200_LONG*

```

DATA: wa_cabecera TYPE ztpp_cotiza_cab.

```

```

DATA: fecha TYPE string.

```

```

DATA: hora TYPE string.

```

```

DATA: lv_vigencia TYPE string.

```

```

DATA: lv_garantia TYPE string.

```

```

DATA: lv_garantia2 TYPE string.

```

```

DATA: lv_plazo TYPE string.

```

```

DATA: lv_meses TYPE i.

```

```

SELECT * FROM ztpp_constant INTO TABLE lv_const_cotiza WHERE modulo = '
PP' AND aplicacion = 'COTIZACION'.

```

```

LOOP AT lv_const_cotiza.
  IF lv_const_cotiza-campo = 'VIGENCIA_COT'.
    lv_vigencia = lv_const_cotiza-valor1.
  ENDIF.
  IF lv_const_cotiza-campo = 'GARANTIA'.
    lv_garantia = lv_const_cotiza-valor1.
    IF lv_garantia < 12.
      IF lv_garantia = 1.
        CONCATENATE lv_garantia 'mes' INTO lv_garantia SEPARATED BY space.
      ELSE.
        CONCATENATE lv_garantia 'meses' INTO lv_garantia SEPARATED BY space.
      ENDIF.
    ELSE.
      lv_garantia2 = lv_garantia DIV 12.
      lv_garantia = lv_garantia MOD 12.
      lv_meses = lv_garantia.
      IF lv_meses IS INITIAL.
        CONCATENATE lv_garantia2 'año(s)' INTO lv_garantia .
      ELSE.
        IF lv_garantia = 1.
          CONCATENATE lv_garantia2 'año(s)' lv_garantia 'mes' INTO lv_garantia SEPARATED BY space.
        ELSE.
          CONCATENATE lv_garantia2 'año(s)' lv_garantia 'meses' INTO lv_garantia SEPARATED BY space.
        ENDIF.
      ENDIF.
    ENDIF.
  ENDIF.
  IF lv_const_cotiza-campo = 'PLAZO_ENT'.
    lv_plazo = lv_const_cotiza-valor1.
  ENDIF.
ENDLOOP.

```

```

* LOOP AT gt_cotiza_outtab_alv . "ADD DB7 VISUALIZ
ZAR CARACT EN COTIZAC COMENT
LOOP AT gt_cotiza_outtab_alv WHERE visualizar NE 'X'. "ADD DB7 VISUALIZ
AR CARACT EN COTIZAC

```

```

MOVE-CORRESPONDING gt_cotiza_outtab_alv TO lt_detalle_cot.
APPEND lt_detalle_cot.
ENDLOOP.

CONCATENATE sy-datum+6(2) sy-datum+4(2) sy-
datum+0(4) INTO fecha SEPARATED BY '/'.
CONCATENATE sy-uzeit+0(2) sy-uzeit+2(2) sy-
uzeit+4(2) INTO hora SEPARATED BY ':'.
* HORA = SY-
DATA: dmpv_indice TYPE sy-tabix.
* DATA: dmpv_indice_aux type sy-tabix.
FIELD-SYMBOLS: <lfs_det> TYPE LINE OF lt_detalle_cot.
SELECT SINGLE * FROM ztpc_cotiza_cab INTO wa_cabecera WHERE num_cot = d
mp200_numcot.
IF sy-subrc = 0.
SELECT SINGLE * FROM ztpc_claves INTO lw_clave WHERE codigo = wa_cabe
cera-cod_ref.
IF wa_cabecera-f_v_cot_d wa_cabecera-f_v_cot_h IS NOT INITIAL.
lv_vigencia = wa_cabecera-f_v_cot_h - wa_cabecera-f_v_cot_d.
ENDIF.
lv_plazo = wa_cabecera-plazo_ent.
LOOP AT gt_cotiza_outtab_alv.
IF gt_cotiza_outtab_alv-titulo = 'CHASIS'.
lv_indice = sy-tabix + 1.
* lv_indice_aux = lv_indice.
LOOP AT lt_detalle_cot ASSIGNING <lfs_det>.
IF sy-tabix >= lv_indice.
<lfs_det>-ord_tit = <lfs_det>-ord_tit + 6.
ENDIF.
ENDLOOP.
lv_detalle_cot-ord_tit = lv_indice.
lv_detalle_cot-subtitulo = 'MARCA CHASIS'.
lv_detalle_cot-valor = lw_clave-mar_cha.
APPEND lv_detalle_cot TO lt_detalle_cot.
CLEAR lv_detalle_cot.

lv_detalle_cot-ord_tit = lv_indice + 1.
lv_detalle_cot-subtitulo = 'MODELO CHASIS'.
lv_detalle_cot-valor = lw_clave-mod_cha.

```



```

APPEND lw_detalle_cot TO lt_detalle_cot.
CLEAR lw_detalle_cot.

lw_detalle_cot-ord_tit = lv_indice + 2.
lw_detalle_cot-subtitulo = 'COMBUSTIBLE'.
lw_detalle_cot-valor = lw_clave-cobust.
APPEND lw_detalle_cot TO lt_detalle_cot.
CLEAR lw_detalle_cot.

lw_detalle_cot-ord_tit = lv_indice + 3.
lw_detalle_cot-subtitulo = 'MOTOR'.
lw_detalle_cot-valor = lw_clave-motor.
APPEND lw_detalle_cot TO lt_detalle_cot.
CLEAR lw_detalle_cot.

lw_detalle_cot-ord_tit = lv_indice + 4.
lw_detalle_cot-subtitulo = 'MODELO MOTOR'.
lw_detalle_cot-valor = lw_clave-mod_motor.
APPEND lw_detalle_cot TO lt_detalle_cot.
CLEAR lw_detalle_cot.

lw_detalle_cot-ord_tit = lv_indice + 5.
lw_detalle_cot-subtitulo = 'FORMULA RODANTE'.
lw_detalle_cot-valor = lw_clave-form_rod.
APPEND lw_detalle_cot TO lt_detalle_cot.
CLEAR lw_detalle_cot.

EXIT.
ENDIF.
ENDLOOP.
SORT lt_detalle_cot BY ord_tit ASCENDING.
DATA: dmpv_xcpdk TYPE knal-xcpdk.
DATA: dmpv_kunnr TYPE knal-kunnr.
lv_kunnr = wa_cabecera-kunnr2.
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
EXPORTING
input = lv_kunnr
IMPORTING
output = lv_kunnr.
SELECT SINGLE xcpdk FROM knal INTO (lv_xcpdk)
WHERE kunnr = lv_kunnr.
IF sy-subrc = 0.

```

```

IF wa_cabecera-kunnr = wa_cabecera-kunnr2 AND lv_xcpdk = 'X'.
    wa_cabecera-kunnr2_1 = wa_cabecera-nombre_cli.
ENDIF.
ENDIF.
* IF wa_cabecera-MAR_CHA wa_cabecera-MOD_CHA IS NOT INITIAL.
* LOOP AT gt_cotiza_outtab_alv.
* MOVE-CORRESPONDING gt_cotiza_outtab_alv TO lt_detalle_cot.
* APPEND lt_detalle_cot.
** IF lt_detalle_cot-titulo = 'CHASIS'.
**
** ENENDIF.
* ENDLLOOP.
* ELSE.
* LOOP AT gt_cotiza_outtab_alv.
* MOVE-CORRESPONDING gt_cotiza_outtab_alv TO lt_detalle_cot.
* APPEND lt_detalle_cot.
* ENDLLOOP.
*
* ENENDIF.
DATA: dmpv_valor1 TYPE ztpp_constant-valor1,
      lv_valor2 TYPE ztpp_constant-valor2,
      lv_subtit TYPE zpp_cotiza_det-subtitulo,
      lv_tit TYPE zpp_cotiza_det-titulo,
      lv_pln TYPE string.
CLEAR: lv_valor1,lv_valor2,lv_subtit,lv_tit,lv_pln.
DATA: lt_planos TYPE STANDARD TABLE OF ztpp_constant WITH HEADER LINE
.
SELECT * FROM ztpp_constant INTO TABLE lt_planos WHERE
modulo = 'PP' AND
aplicacion = 'COTIZACION' AND
programa = 'IMAGEN_PLANO' AND
campo = 'IMAGEN_PLANO' AND
valor1 = wa_cabecera-matnr.
IF sy-subrc = 0.
LOOP AT lt_planos.
READ TABLE gt_cotiza_outtab_alv WITH KEY valor = lt_planos-
valor2.
IF sy-subrc = 0.
READ TABLE gt_cotiza_outtab_alv WITH KEY valor = lt_planos-
valor3.
IF sy-subrc = 0.
lv_pln = lt_planos-valor4.

```

```

        EXIT.
    ENDIF.
ENDIF.
* LOOP AT gt_cotiza_outtab_alv.
*     if gt_cotiza_outtab_alv-titulo is not initial.
*         lv_tit = gt_cotiza_outtab_alv-titulo.
*     endif.
*     if gt_cotiza_outtab_alv-subtitulo is not initial.
*         lv_subtit = gt_cotiza_outtab_alv-subtitulo.
*     endif.
*     IF lv_tit = 'CARROCERIA' AND lv_subtit = 'CANTIDAD DE ASIENTOS'.
*         if gt_cotiza_outtab_alv-valor is not initial.
*             lv_valor1 = gt_cotiza_outtab_alv-valor.
*         endif.
*     ENDIF.
* ENDLOOP.
ENDIF.

```

```

CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
EXPORTING
    formname           = 'ZPPP_SF_COTIZACION'
* VARIANT              = ' '
* DIRECT_CALL         = ' '
IMPORTING
    fm_name           = lv_fmname
EXCEPTIONS
    no_form           = 1
    no_function_module = 2
    OTHERS            = 3.
.

```

```

IF sy-subrc <> 0.
* Implement suitable error handling here
ENDIF.

CLEAR gs_otfdata.
IF imprime = 'X'.
    CLEAR gs_ctrlop.
*     ls_control-no_dialog = 'X'.

```

```

gs_ctrlop-device      = 'PRINTER'.
gs_ctrlop-preview    = 'X'.
CLEAR gs_outopt.
*  ls_output-tdnoprev = 'X'.
gs_outopt-tddest     = 'LOCL'.
ELSE.
CLEAR gs_ctrlop.
CLEAR gs_outopt.

gs_ctrlop-no_dialog = 'X'.
gs_outopt-tddest    = 'LOCL'.
gs_outopt-tdlifetime = '1'.
gs_outopt-tdnewid   = 'X'.
gs_outopt-tdarmod   = '1'.
gs_outopt-tdcover   = ' '.
gs_outopt-tdprinter = 'SWIN'.
gs_outopt-tdcopies  = 001.
ENDIF.

DATA: nombre TYPE string,
      ruc     TYPE string.

nombre = dmp200_l_kunnr.
ruc     = dmp300_c_dni.

CALL FUNCTION lv_fmname
EXPORTING
control_parameters = gs_ctrlop
output_options     = gs_outopt
wa_cabecera       = wa_cabecera
fecha             = fecha
hora              = hora
vigencia          = lv_vigencia
garantia          = lv_garantia
plazo             = lv_plazo
imagen            = gv_imagen
razon_social      = nombre
ruc               = ruc
pln               = lv_pln
*  USER_SETTINGS  = 'X'
*  p_aufnr        = p_aufnr
*  p_chasis       = v_chasis

```

```

*      p_nombre          = v_nombre
*      p_ltxal          = v_ltxal
*      p_mda            = p_mda
IMPORTING
*      DOCUMENT_OUTPUT_INFO      =
      job_output_info    = gs_otfdata
*      JOB_OUTPUT_OPTIONS =
TABLES
      t_detalle_cot      = lt_detalle_cot
EXCEPTIONS
      formatting_error   = 1
      internal_error     = 2
      send_error         = 3
      user_canceled     = 4
      OTHERS             = 5.
IF sy-subrc <> 0.
* Implement suitable error handling here
ENDIF.
IF imprime = ''.
      PERFORM descargar_pdf USING gs_otfdata .
ENDIF.
ENDIF.
ENDFORM.
*
FORM descargar_pdf USING gs_otfdata TYPE ssfcrescl.

      DATA: gv_form_name      TYPE rs38l_fnam,
            gv_bin_x°ilesize TYPE i, " Store the file size
            gv_pos            TYPE i,
            gv_len            TYPE i,
            gv_tab_lines     TYPE i.
*****Constants *****
      DATA : gc_text(11) TYPE c VALUE 'Form Output',
            gc_tst(3)   TYPE c VALUE 'TST'.
*****Work Area Declarations *****
      DATA: gw_docdata TYPE sodocchg1,
            gw_ctrlop  TYPE ssfctrlop, " Smart Forms: Control structure
            gw_outopt  TYPE ssfcompop, *          gs_otfdata type ssfcrescl,
" Smart Forms: Return value at end of form printing

```

```

        gs_reclist TYPE somlrecil, *           " Workarea for SAP Script
Text Lines
        gs_ubjbin  TYPE solistil,           gs_ubjpack TYPE sopcklstil. "
SAPoffice: Description of Imported Object Components
*****Internal tables Declarations *****
DATA:  gt_reclist TYPE TABLE OF somlrecil, "
*      gt_pdf_tab type table of tline, " SAPscript: Text Lines
        gt_utl1f   TYPE TABLE OF itcoo, "
        gt_ubljbin TYPE TABLE OF solistil, " SAPoffice: Single List wit
h Column Length 255
        gt_objpack TYPE TABLE OF sopcklstil. "
* data p_sender TYPE somlrecil-receiver value 'alertas@modasa.com.pe'.
* data gd_sender_tyYype TYPE soextrecil-adr_typ.
* data w_

*
* call function 'SSF_FUNCTION_MODULE_NAME'
*   exporting
*     formname           = 'ZOSFISF_MAIL_RETENCION_IGV'

*  NOTA Si se desea que SMARTFORMS retorne OTF habilitar OTF y las lineas
en comentario lineas abajo despues del
* envio de email.
*
* gs_outopt-tdlifetime = '1'.
* gs_outopt-tdnewid = 'X'.
* gs_outopt-tdarmod = '1'.
* gs_outopt-tdcover = ' '.
* gs_outopt-tdprinter = 'SWIN'.
* gs_outopt-tdcopies = 001.
**the OTFDAT
*
*
* call function gv_form_name ''
*   wi_total           = w_total
*   wi_totall         = w_totall
*   wi_total2         = w_total2
*   wi_texto          = w_texto
*   wi_bukrs          = wa_cabeceral-bukrs

```

```

*      wi_gjahr          = wa_cabeceral-gjahr
*      wi_belnr         = wa_cabeceral-belnr
*      wi_xblnr        = wa_cabeceral-ctnumber
*      wi_name1         = wa_cabeceral-name1
*      wi_name2         = wa_cabeceral-name2
*      wi_stras         = wa_cabeceral-stras
*      wi_ort01         = wa_cabeceral-ort01
*      wi_ort02         = wa_cabeceral-ort02
*      wi_stcd1         = wa_cabeceral-stcd1
*      wi_bldat         = wa_cabeceral-bldat
*      wi_waers         = wa_cabeceral-waers           "gw_waers01
*      wi_kursf         = wa_cabeceral-kursf
*      wi_wrbtr         = w_wrbtr
*      wi_lifnr         = wa_cabecera-lifnr

```

```

*      importing
*      job_output_info  = gs_otfdata
*      tables
*      gt_detalle       = t_detalle2
*      exceptions
*      formatting_error = 1
*      internal_error   = 2
*      send_error       = 3
*      user_canceled    = 4.

```

```

*      if sy-subrc ne 0 .
*      message id sy-msgid type sy-msgty number sy-msgno
*      with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
*      endif.

```

```

*****
*****

```

```

"CON SPOOL
TABLES tsp01.
DATA t_spool TYPE tsfspoolid.
DATA w_spool TYPE rspoid.
DATA i_otf.
DATA: pbyte TYPE i.
DATA: t_pdf TYPE STANDARD TABLE OF tline .
DATA: i_pdf TYPE tline.
DATA gd_buffer TYPE string.
DATA it_mess_att TYPE solist11 OCCURS 0 WITH HEADER LINE.
DATA gd_recsiz TYPE i.
DATA p_odescr TYPE sodocchg1-obj_desr VALUE 'descripcion A'.

```

```

DATA: gd_subject TYPE sodocchg1l-obj_descr.
DATA p_adescr TYPE so_obj_nam VALUE 'Descripcion B'.
DATA gd_attachment_desc TYPE so_obj_nam.
DATA v_spool_id TYPE rspoid.
DATA: client1 TYPE tst01-dclient.
DATA nae TYPE tst01-dname.
DATA pdfsplid TYPE tsp01-rqident.
DATA jobnme TYPE tbtccjob-jobname.
DATA jobcnt TYPE tbtccjob-jobcount.
DATA type TYPE rststype-type.
DATA objtype TYPE rststype-type.
DATA: t_receivers TYPE somlrecil OCCURS 0 WITH HEADER LINE,
      vigenciai TYPE string,
      vigenciaf TYPE string,
      fechai TYPE sy-datum,
      fechaf TYPE sy-datum.
DATA: p_file TYPE string VALUE 'C:\'.

```

```

t_spool = gs_otfdata-spoolids.
READ TABLE t_spool INTO w_spool INDEX 1.
IF sy-subrc = 0.
  SELECT SINGLE * FROM tsp01 WHERE rqident = w_spool.
  client = tsp01-rqclient.
  name = tsp01-rqolname.
  v_spool_id = w_spool.

```

```

IF sy-subrc = 0.
  FREE: t_pdf.

```

```

ENDIF.

```

```

LOOP AT t_pdf INTO i_pdf.
endloop

```

```

DESCRIBE TABLE it_mess_att LINES gd_recsz.
IF gd_recsz > 0 .

```



```

DATA ld_spool_nr TYPE tsp01_sp0r-rqid_char.
DATA w_cnt TYPE i.
DATA gd_spool_nr TYPE tsp01-rqidident.
gd_spool_nr = v_spool_id.
ld_spool_nr = gd_spool_nr.

* Fill the document data.
DATA w_doc_data TYPE sodocchg1.
DATA t_attachment TYPE solist1l OCCURS 0 WITH HEADER LINE.
w_doc_data-doc_size = 1.

CLEAR w_doc_data..

*   call function 'ISP_GET_MONTH_NAME'
*   exporting
*   date           = fechai
*   language       = sy-langu
**  MONTH_NUMBER  = '00'
*   importing
**  LANGU_BACK    =
*   longtext       = v_mes1.
*
*   data: aux1 type string,
*         aux2 type string.
*   concatenate p_gjahr ') 'into aux1.
*   concatenate 'Comprobante de Retención (' v_mes1 into aux2.
*   concatenate aux2 aux1 into separated by space.
*
*
*   data t_packing_list TYPE sopck1st1l occurs 0 with header line.
*
*
*
*   call function 'FKK_LAST_DAY_OF_MONTH'

```

```

*      exporting
*      day_in                = fechai
*      importing
*      last_day_of_month     = fechaf
**     EXCEPTIONS
**     DAY_IN_NO_DATE       = 1
**     OTHERS               = 2
*
*      .
*      concatenate fechai+0(4) fechai+4(2) fechai+6(2) into vigenciai.
*      concatenate fechaf+0(4) fechaf+4(2) fechaf+6(2) into vigenciaf.
*
*
*
*      clear t_receivers.
*      refresh t_receivers.
*
*      IF sy-subrc = 0.
*
*      loop at t_correo2 assigning <fv_persona>.
*          refresh it_mess_bod.
*
*          clear it_mess_bod.
*          concatenate 'Estimado ' <fv_persona>-
namev   into it_mess_bod separated by space.
= 'Adjuntamos el certificado de retención con vigencia del 01/05/2014 al
hasta el 31/05/2014'.
*          concatenate 'Adjuntamos el certificado de retención con vigencia d
el' vigenciai 'hasta el ' vigenciaf into it_mess_bod separated by space.
*
*          it_mess_bod = 'De tener alguna observación sírvase contactar con e
l area de Tesorería, cuyo nombre y teléfono se indica en el documento.'.
*
*          it_mess_bod = 'Motores Diesel Andinos S.A.'.
= 'www.modasa.com.pe'.
*          append it_mess_bod.
*
*      'PDF'.
*      clear aux1.
*      concatenate p_gjahr ').pdf' into aux1.
*      concatenate 'Comprobante de Retención (' v_mes1 into aux2.

```

```

*      concatenate aux2 aux1 into t_packing_list-
obj_descr separated by space.
*
*      t_packing_list-obj_name = 'ATTACH_NAME'.
*      t_packing_list-doc_size = t_packing_list-body_num * 255.
*      append t_packing_list.
*
*      refresh t_receivers.
*      clear t_receivers.
*      if <fv_persona>-correo is not initial.
*          concatenate <fv_persona>-correo ' ' into w_email-valor1.
**      CONCATENATE 'csarazu@iterpconsulting.com' ' ' INTO w_email-valor1.

*      endif.
*      endloop.

*clear p_file.

*prov = wa_cabeceral-wt_acco.
*
*call function 'CONVERSION_EXIT_ALPHA_OUTPUT'
* exporting
* input = prov
* importing
* output = prov.
*condense wa_cabeceral-ctnumber no-gaps.
*condense prov no-gaps.
*      add 1 to contador2.

      initial_folder = 'C:\Documents and Settings\All Users\Escrit
orio'
      CHANGING
      selected_folder = p_file.
*      EXCEPTIONS:
*      cntl_error = 1
*      error_no_gui = 2
*      not_supported_by_gui = 3.
CONCATENATE p_file '\COT_' dmp200_numcot '.pdf' INTO p_file.

      ENDIF.

```

```

ENDIF.

ENDFORM.                "call_smartform2

*&-----*
*&      Form  GUARDA_ESTRATEGIAS
*&-----*
*
FORM guarda_estrategias TABLES lt_liberacion STRUCTURE ztpp_liberacion USING lv_flag_solicitud.
  DATA: gv_envia_correo.
  DATA: lt_constantes TYPE STANDARD TABLE OF ztpp_constant WITH HEADER LINE.
  DATA: "lt_liberacion      TYPE STANDARD TABLE OF ztpp_liberacion WITH HEADER LINE,
         lt_liberacion_aux TYPE STANDARD TABLE OF ztpp_liberacion WITH HEADER LINE.
  DATA: name1 TYPE adrp-mc_namefir,
         name2 TYPE adrp-mc_namelas.
  DATA: t_margenes TYPE STANDARD TABLE OF ztpp_constant WITH HEADER LINE.
  REFRESH lt_liberacion.
  DATA: marg_aux TYPE ze_pp_pv_netto,
         marg_aux2 TYPE ze_pp_pv_netto..

IF dmp200_land1 = 'PE'.
  SELECT * INTO TABLE t_margenes FROM ztpp_constant
    WHERE modulo = 'PP' AND
           aplicacion = 'COTIZACION' AND
           programa = 'MARGEN_GANANCIA' AND
           valor4 = dmp200_land1.

ELSE.
  SELECT * INTO TABLE t_margenes FROM ztpp_constant
    WHERE modulo = 'PP' AND
           aplicacion = 'COTIZACION' AND
           programa = 'MARGEN_GANANCIA' AND
           valor4 = ''.

ENDIF.

```

```

REFRESH lt_constantes.
SELECT * INTO TABLE lt_constantes FROM ztpp_constant
WHERE modulo = 'PP' AND
      aplicacion = 'LIBERACION' AND
      programa = 'COTIZACION' .

CALL FUNCTION 'POPUP_TO_CONFIRM'
EXPORTING
*   TITLEBAR           = ' '
*   DIAGNOSE_OBJECT    = ' '
      text_question     = 'DESEA ENVIAR ALERTA PARA LIBERACION DE COT
IZACION?'
      text_button_1     = 'OK'
      icon_button_1     = 'ICON_CHECKED'
      text_button_2     = 'CANCEL'
      icon_button_2     = 'ICON_INCOMPLETE'
      default_button    = ''
      display_cancel_button = ''
IMPORTING
      answer            = answer.
IF answer = 1.
      gv_envia_correo = 'X'.
ENDIF.
*SELECT * into TABLE lt_liberacion_aux from ztpp_liberacion
*   where NUM_COT = dmp200_numcot.
*   marg_aux = gv_marg_ganancha_h + 5 / 100.
*   marg_aux2 = gv_marg_gananc_h + 5 / 100.

* LOOP AT t_margenes WHERE campo = 'MRG_CHASIS'.

marg_aux = gv_margen_chas_inf ." + 5 / 100.
marg_aux2 = gv_margen_chas_sup." + 5 / 100.
IF dmp200_particip_chasis_f <= marg_aux2 AND dmp200_particip_chasis_f
>= marg_aux.
      READ TABLE t_margenes WITH KEY campo = 'MRG_CHAIS'.
      CLEAR lt_constantes.
      lt_liberacion-num_cot = dmp200_numcot.
      lt_liberacion-gr_libera = 'EST_LIBERA_CHAS'.

```

```

READ TABLE lt_constantes WITH KEY campo = 'EST_LIBERA_CHAS'.
lt_liberacion-descrip = lt_constantes-valor2.
lt_liberacion-priorida = lt_constantes-valor3.
lt_liberacion-usuario = lt_constantes-valor1.
CLEAR: name1, name2.
SELECT SINGLE a~mc_namefir a~mc_namelas INTO (name1, name2) FROM adrp
      WHERE bname = lt_liberacion-usuario.
ENDIF.
APPEND lt_liberacion.
CLEAR lt_liberacion.
IF gv_envia_correo = 'X'.
      PERFORM envia_correo USING dmp200_numcot 'EST_LIBERA_CHAS'.
ENDIF.
*      EXIT.
ENDIF.

*      ENDLOOP.

```

```

LOOP AT t_margenes WHERE campo = 'MRG_GAIN'.

      marg_aux = t_margenes-valor1." + 5 / 100.
      marg_aux2 = t_margenes-valor2." + 5 / 100.
      IF dmp200_particip_carroc_f <= marg_aux2 AND dmp200_particip_carroc
_f > marg_aux.
            CLEAR lt_constantes.
            lt_liberacion-num_cot = dmp200_numcot.
            lt_liberacion-gr_libera = 'EST_LIBERA_MARG'.
            lt_liberacion-usuario = t_margenes-valor3.

            CLEAR: name1, name2.
            SELECT SINGLE a~mc_namefir a~mc_namelas INTO (name1, name2) FROM ad
rp
                  WHERE bname = lt_liberacion-usuario.

            ENDIF.
            READ TABLE lt_constantes WITH KEY campo = 'EST_LIBERA_MARG'.
            lt_liberacion-descrip = lt_constantes-valor2.

```

```

lt_liberacion-priorida = lt_constantes-valor3.

APPEND lt_liberacion.
CLEAR lt_liberacion.
IF gv_envia_correo = 'X'.
    PERFORM envia_correo USING dmp200_numcot 'EST_LIBERA_MARG'.
ENDIF.
*     EXIT.
ENDIF.

ENDLOOP.

*
* IF dmp200_particip_chasis_f < gv_marg_ganancha_1 OR dmp200_particip_ch
asis_f > marg_aux .
*     LOOP AT lt_constantes WHERE campo = 'EST_LIBERA_MARG' AND valor4 = '
X'.
**     READ TABLE lt_liberacion_aux
*
*     lt_liberacion-num_cot = dmp200_numcot.
*     lt_liberacion-gr_libera = lt_constantes-campo.
*     lt_liberacion-usuario = lt_constantes-valor1.
*
*     CLEAR: name1, name2.
*     SELECT SINGLE a~mc_namefir a~mc_name1as INTO (name1, name2) FROM a
drp
*
*     WHERE bname = lt_liberacion-usuario.
*
*     ENDIF.
*     lt_liberacion-descrip = lt_constantes-valor2.
*     lt_liberacion-priorida = lt_constantes-valor3.
*     APPEND lt_liberacion.
*     CLEAR lt_liberacion.
*     ENDLOOP.
*     PERFORM envia_correo USING dmp200_numcot 'EST_LIBERA_MARG'.
*
* ELSEIF dmp200_particip_carroc_f < gv_marg_gananc_1 OR dmp200_particip
_carroc_f > marg_aux2.
*     LOOP AT lt_constantes WHERE campo = 'EST_LIBERA_MARG' AND valor4 = '

```

```

X'..
*
*   lt_liberacion-num_cot   = dmp200_numcot.
*   lt_liberacion-gr_libera = lt_constantes-campo.
*   lt_liberacion-usuario  = lt_constantes-valor1.
*   CLEAR: name1, name2.
*   SELECT SINGLE a~mc_namefir a~mc_namelas INTO (name1, name2) FROM a
drp
*
*       WHERE bname = lt_liberacion-usuario.
*
*   ENDIF.
*   lt_liberacion-descrip   = lt_constantes-valor2.
*   lt_liberacion-priorida  = lt_constantes-valor3.
*   APPEND lt_liberacion.
*   CLEAR lt_liberacion.
*   ENDLOOP.
*   PERFORM envia_correo USING dmp200_numcot 'EST_LIBERA_MARG'.
*   ENDIF.
*
* LOOP AT lt_constantes WHERE campo = 'EST_LIBERA_GRAL' AND valor4 = 'X'
..
*   lt_liberacion-num_cot   = dmp200_numcot.
*   lt_liberacion-gr_libera = lt_constantes-campo.
*   lt_liberacion-usuario  = lt_constantes-valor1.
*   CLEAR: name1, name2.
*   SELECT SINGLE a~mc_namefir a~mc_namelas INTO (name1, name2) FROM adr
*       WHERE bname = lt_liberacion-usuario.
*   ENDIF.
*   lt_liberacion-descrip   = lt_constantes-valor2.
*   lt_liberacion-priorida  = lt_constantes-valor3.
*   APPEND lt_liberacion.
*   CLEAR lt_liberacion.
*   ENDLOOP.
*   perFORM envia_correo USING p_numcot 'EST_LIBERA_MARG'.

```



```
"LONGITUDES COTIZACION
```

```
"LONGITUDES COTIZACION
```

```
IF lv_flag_solicitud = 'X'.  
  LOOP AT lt_constantes WHERE campo = 'EST_LIBERA_NUEV' AND valor4 = 'X'  
'..  
    lt_liberacion-num_cot    = dmp200_numcot.  
    lt_liberacion-gr_libera  = lt_constantes-campo.  
    lt_liberacion-usuario    = lt_constantes-valor1.  
    CLEAR: name1, name2.  
    SELECT SINGLE a~mc_namefir a~mc_namelas INTO (name1, name2) FROM ad  
rp AS aa  
      INNER JOIN usr21 AS u ON u~persnumber = a~persnumber  
      WHERE bname = lt_liberacion-usuario.  
    IF sy-subrc = 0.  
      CONCATENATE name1 name2 INTO lt_liberacion-  
nombre SEPARATED BY space..  
    ENDIF.  
    lt_liberacion-descrip    = lt_constantes-valor2.  
    lt_liberacion-priorida   = lt_constantes-valor3.  
    APPEND lt_liberacion.  
    CLEAR lt_liberacion.  
  ENDLOOP.  
IF gv_envia_correo = 'X'.  
  PERFORM envia_correo USING dmp200_numcot 'EST_LIBERA_NUEV'.  
ENDIF.  
ENDIF.  
DELETE FROM ztpp_liberacion WHERE num_cot    = dmp200_numcot.  
MODIFY   ztpp_liberacion FROM TABLE lt_liberacion[].  
  
IF lt_liberacion[] IS INITIAL.  
  UPDATE ztpp_cotiza_cab SET aceptado = 'X'  
  WHERE num_cot = dmp200_numcot.  
  
  IF dmp200_confirmado = 'X'.  
  
  ELSE.  
    dmp200_aceptado      = 'X'.  
    mp200_estado_cotiza = 'Liberado'.  
  ENDIF.  
ELSE.  
  CLEAR dmp200_confirmado .
```

```

UPDATE ztpp_cotiza_cab SET aceptado = ''
WHERE num_cot = dmp200_numcot.
dmp200_aceptado      = ''.
mp200_estado_cotiza = 'Bloqueado'.

ENDIF.

ENDFORM.

FORM valor_nuevo TABLES t_plantillas1_aux STRUCTURE wa_plantilla1
                        t_plantillas2_aux STRUCTURE wa_plantilla2
                        USING      p_e_fieldname p_e_value
                        s_row_no STRUCTURE s_roid.
FIELD-SYMBOLS: <fv_cotiza> TYPE LINE OF gt_cotiza_outtab,
               <fs_cal>    TYPE any,
               <fs_val>    TYPE any,
               <config>    TYPE any,
               <fs_val2>   TYPE any.
DATA: dmpv_restriccion TYPE c.
CLEAR lv_restriccion.
* DATA: t_valor TYPE STANDARD TABLE OF ty_valor.
REFRESH t_valor.
DATA: w_valor      TYPE t_valor,
      w_config     TYPE t_config,
      lv_cotiza2   TYPE gt_cotiza_outtab,
      nombre       TYPE string,
      nombre3      TYPE string,
      nombre2      TYPE string
      .

DATA: dmpv_cant    TYPE i,
      lv_cant2     TYPE i,
      lv_index     TYPE i.

DATA: l_plantillas1 TYPE LINE OF t_plantillas1,
      l_plantillas2 TYPE LINE OF t_plantillas2,
      l_compara.
* DATA: t_plantillas1_aux TYPE STANDARD TABLE OF ztpp_valor1,
*       t_plantillas2_aux TYPE STANDARD TABLE OF ztpp_valor2.

DATA: lv_contador(3) TYPE n VALUE '000'.

```

```
DATA: gt_cotiza_out_aux TYPE STANNRDAR TABLE OF gty_cotiza_outb WITH  
HEADER LINE.
```

```
gv_codigo = dmp200_producto.
```

```
* CALL METHOD grid_cotiza->check_changed_data.
```

```
REFRESH: t_valor.
```

```
TYPES: BEGIN OF ty_mach.
```

```
INCLUDE STRUCTURE ztpp_valor1.
```

```
TYPES: col080 (50),
```

```
col081 (50),
```

```
col082 (50),
```

```
col083 (50),
```

```
col084 (50),
```

```
col085 (50),
```

```
col086 (50),
```

```
col087 (50),
```

```
col088 (50),
```

```
col089 (50),
```

```
col090 (50),
```

```
col091 (50),
```

```
col092 (50),
```

```
col093 (50),
```

```
col094 (50),
```

```
col095 (50),
```

```
col096 (50),
```

```
col097 (50),
```

```
col098 (50),
```

```
col099 (50),
```

```
col100 (50),
```

```
col101 (50),
```

```
col102 (50),
```

```
col103 (50),
```

```
col104 (50),
```

```
col105 (50),
```

```
col106 (50),
```

```
col107 (50),
```

```
col108 (50),
```

```
col109 (50),
```

```
col110 (50),
```

```
col111 (50),
```

```
col112 (50),
```

```
col113 (50),
```

col114 (50),
col115 (50),
col116 (50),
col117 (50),
col118 (50),
col119 (50),
col120 (50),
col121 (50),
col122 (50),
col123 (50),
col124 (50),
col125 (50),
col126 (50),
col127 (50),
col128 (50),
col129 (50),
col130 (50),
col131 (50),
col132 (50),
col133 (50),
col134 (50),
col135 (50),
col136 (50),
col137 (50),
col138 (50),
col139 (50),
col140 (50),

END OF ty_mach.

DATA: lt_configu_2 TYPE STANNDARD TABLE OF ty_restr .

DATA: lt_rest1 TYPE STANNDARD TABLE OF ztpp_restr1 .

DATA: lt_rest2 TYPE STANNDARD TABLE OF ztpp_restr2 .

DATA: wa_configu_2 TYPE ty_restr .

DATA: wa_configu_2_aux TYPE ty_restr .

FIELD-SYMBOLS: <fis_configu_2> TYPE LINE OF lt_configu_2.

DATA: wa_mach TYPE ty_mach .

DATA: t_mach TYPE STANDARD TABLE OF ty_mach WITH HEADER LINE.

lt_rest1[] = t_rest1[].

lt_rest2[] = t_rest2[].

```

lt_configu_2[] = t_configu_2[].
LOOP AT lt_rest1.
  CLEAR: l_compara,lv_contador.
  READ TABLE lt_rest2 WITH KEY fila = lt_rest1-fila BINARY SEARCH.
  MOVE-CORRESPONDING lt_rest1 TO t_mach.
  MOVE-CORRESPONDING lt_rest2 TO t_mach.
  APPEND t_mach.

ENDLOOP.

LOOP AT lt_configu_2 ASSIGNING <fs_configu_2>.
  READ TABLE t_config WITH KEY titulo = <fs_configu_2>-
titulo subtit = <fs_configu_2>-subtit
                                caract = <fs_configu_2>-
caract subcar = <fs_configu_2>-subcar.
  IF sy-subrc = 0.
    <fs_configu_2>-columna_val = t_config-columna.
  ENDIF.
ENDLOOP.

"ADD DB7 VISUALIZAR CARACT EN COTIZAC INI
READ TABLE gt_cotiza_outtab_alv_x INDEX p_es_row_no-row_id.
IF sy-subrc = 0.
  p_es_row_no-row_id = gt_cotiza_outtab_alv_x-ord_tit.
ENDIF.

"ADD DB7 VISUALIZAR CARACT EN COTIZAC END

READ TABLE t_cotiza_outtab INTO l_cotiza2 INDEX p_s_rw_no-rw_id.
IF lw_cotiza2-subtitulo <> 'MARCA-MODELO-CHASIS' AND
lw_cotiza2-subtitulo <> 'MARCA-MODELO-MOTOR' AND
lw_cotiza2-subtitulo <> 'COMBUSTIBLE' AND
lw_cotiza2-subtitulo <> 'FORMULA RODANTE'.

  READ TABLE t_config INTO w_config WITH KEY titulo = lw_cotiza2-
titulo
                                subtit = lw_cotiza2-
subtitulo
                                caract = lw_cotiza2-
caracterica
                                subcar = lw_cotiza2-
subcaracteristica.

```

```

IF sy-subrc = 0.
    "CONSULTA SI HAY ALGUNA RESTRICCIÓN
    SORT lt_configu_2 BY columna_val ASCENDING.
    LOOP AT lt_configu_2 INTO wa_configu_2 WHERE columna_val <= w_config-
columna.
    ENDLOOP.
    SORT lt_configu_2 BY columna ASCENDING.
    IF wa_configu_2 IS NOT INITIAL.
        DATA: colum_rest(3) TYPE n.
        LOOP AT lt_configu_2 INTO wa_configu_2_aux.
            ADD 1 TO colum_rest.
*****OBTENER VALOR DE COLUMNA DE LA PLANTILLA*****
*****
        IF wa_configu_2_aux-columna_val < 80.
            CONCATENATE 'wa_plantilla1-col' wa_configu_2_aux-
columna_val INTO nombre.
        ELSE.
            CONCATENATE 'wa_plantilla2-col' wa_configu_2_aux-
columna_val INTO nombre.
        ENDIF.
        ASSIGN (nombre) TO <fs_val>.
*****OBTENER VALOR DE COLUMNA DE LA PLANTILLA*****
*****
*****ELIMINAR RESTRICCIONES SEGUN COLUMNA*****
        IF wa_configu_2_aux-columna_val <> w_config-columna.
            CASE wa_configu_2_aux-columna.
                WHEN '001'.
                    DELETE t_mach WHERE col001 <> <fs_val>.
                WHEN '002'.
                    DELETE t_mach WHERE col002 <> <fs_val>.
                WHEN '003'.
                    DELETE t_mach WHERE col003 <> <fs_val>.
                WHEN '004'.
                    DELETE t_mach WHERE col004 <> <fs_val>.
                WHEN '005'.
                    DELETE t_mach WHERE col005 <> <fs_val>.
                WHEN '006'. DELETE t_mach WHERE col006 <> <fs_val>.
                WHEN '007'. DELETE t_mach WHERE col007 <> <fs_val>.
                WHEN '008'. DELETE t_mach WHERE col008 <> <fs_val>.
                WHEN '009'. DELETE t_mach WHERE col009 <> <fs_val>.

```


WHEN '052'. DELETE t_mach WHERE col052 <> <fs_val>.
WHEN '053'. DELETE t_mach WHERE col053 <> <fs_val>.
WHEN '054'. DELETE t_mach WHERE col054 <> <fs_val>.
WHEN '055'. DELETE t_mach WHERE col055 <> <fs_val>.
WHEN '056'. DELETE t_mach WHERE col056 <> <fs_val>.
WHEN '057'. DELETE t_mach WHERE col057 <> <fs_val>.
WHEN '058'. DELETE t_mach WHERE col058 <> <fs_val>.
WHEN '059'. DELETE t_mach WHERE col059 <> <fs_val>.
WHEN '060'. DELETE t_mach WHERE col060 <> <fs_val>.
WHEN '061'. DELETE t_mach WHERE col061 <> <fs_val>.
WHEN '062'. DELETE t_mach WHERE col062 <> <fs_val>.
WHEN '063'. DELETE t_mach WHERE col063 <> <fs_val>.
WHEN '064'. DELETE t_mach WHERE col064 <> <fs_val>.
WHEN '065'. DELETE t_mach WHERE col065 <> <fs_val>.
WHEN '066'. DELETE t_mach WHERE col066 <> <fs_val>.
WHEN '067'. DELETE t_mach WHERE col067 <> <fs_val>.
WHEN '068'. DELETE t_mach WHERE col068 <> <fs_val>.
WHEN '069'. DELETE t_mach WHERE col069 <> <fs_val>.
WHEN '070'. DELETE t_mach WHERE col070 <> <fs_val>.
WHEN '071'. DELETE t_mach WHERE col071 <> <fs_val>.
WHEN '072'. DELETE t_mach WHERE col072 <> <fs_val>.
WHEN '073'. DELETE t_mach WHERE col073 <> <fs_val>.
WHEN '074'. DELETE t_mach WHERE col074 <> <fs_val>.
WHEN '075'. DELETE t_mach WHERE col075 <> <fs_val>.
WHEN '076'. DELETE t_mach WHERE col076 <> <fs_val>.
WHEN '077'. DELETE t_mach WHERE col077 <> <fs_val>.
WHEN '078'. DELETE t_mach WHERE col078 <> <fs_val>.
WHEN '079'. DELETE t_mach WHERE col079 <> <fs_val>.
WHEN '080'. DELETE t_mach WHERE col080 <> <fs_val>.
WHEN '081'. DELETE t_mach WHERE col081 <> <fs_val>.
WHEN '082'. DELETE t_mach WHERE col082 <> <fs_val>.
WHEN '083'. DELETE t_mach WHERE col083 <> <fs_val>.
WHEN '084'. DELETE t_mach WHERE col084 <> <fs_val>.
WHEN '085'. DELETE t_mach WHERE col085 <> <fs_val>.
WHEN '086'. DELETE t_mach WHERE col086 <> <fs_val>.
WHEN '087'. DELETE t_mach WHERE col087 <> <fs_val>.
WHEN '088'. DELETE t_mach WHERE col088 <> <fs_val>.
WHEN '089'. DELETE t_mach WHERE col089 <> <fs_val>.
WHEN '090'. DELETE t_mach WHERE col090 <> <fs_val>.
WHEN '091'. DELETE t_mach WHERE col091 <> <fs_val>.
WHEN '092'. DELETE t_mach WHERE col092 <> <fs_val>.
WHEN '093'. DELETE t_mach WHERE col093 <> <fs_val>.

WHEN '094'. DELETE t_mach WHERE col094 <> <fs_val>.
WHEN '095'. DELETE t_mach WHERE col095 <> <fs_val>.
WHEN '096'. DELETE t_mach WHERE col096 <> <fs_val>.
WHEN '097'. DELETE t_mach WHERE col097 <> <fs_val>.
WHEN '098'. DELETE t_mach WHERE col098 <> <fs_val>.
WHEN '099'. DELETE t_mach WHERE col099 <> <fs_val>.
WHEN '100'. DELETE t_mach WHERE col100 <> <fs_val>.
WHEN '101'. DELETE t_mach WHERE col101 <> <fs_val>.
WHEN '102'. DELETE t_mach WHERE col102 <> <fs_val>.
WHEN '103'. DELETE t_mach WHERE col103 <> <fs_val>.
WHEN '104'. DELETE t_mach WHERE col104 <> <fs_val>.
WHEN '105'. DELETE t_mach WHERE col105 <> <fs_val>.
WHEN '106'. DELETE t_mach WHERE col106 <> <fs_val>.
WHEN '107'. DELETE t_mach WHERE col107 <> <fs_val>.
WHEN '108'. DELETE t_mach WHERE col108 <> <fs_val>.
WHEN '109'. DELETE t_mach WHERE col109 <> <fs_val>.
WHEN '110'. DELETE t_mach WHERE col110 <> <fs_val>.
WHEN '111'. DELETE t_mach WHERE col111 <> <fs_val>.
WHEN '112'. DELETE t_mach WHERE col112 <> <fs_val>.
WHEN '113'. DELETE t_mach WHERE col113 <> <fs_val>.
WHEN '114'. DELETE t_mach WHERE col114 <> <fs_val>.
WHEN '115'. DELETE t_mach WHERE col115 <> <fs_val>.
WHEN '116'. DELETE t_mach WHERE col116 <> <fs_val>.
WHEN '117'. DELETE t_mach WHERE col117 <> <fs_val>.
WHEN '118'. DELETE t_mach WHERE col118 <> <fs_val>.
WHEN '119'. DELETE t_mach WHERE col119 <> <fs_val>.
WHEN '120'. DELETE t_mach WHERE col120 <> <fs_val>.
WHEN '121'. DELETE t_mach WHERE col121 <> <fs_val>.
WHEN '122'. DELETE t_mach WHERE col122 <> <fs_val>.
WHEN '123'. DELETE t_mach WHERE col123 <> <fs_val>.
WHEN '124'. DELETE t_mach WHERE col124 <> <fs_val>.
WHEN '125'. DELETE t_mach WHERE col125 <> <fs_val>.
WHEN '126'. DELETE t_mach WHERE col126 <> <fs_val>.
WHEN '127'. DELETE t_mach WHERE col127 <> <fs_val>.
WHEN '128'. DELETE t_mach WHERE col128 <> <fs_val>.
WHEN '129'. DELETE t_mach WHERE col129 <> <fs_val>.
WHEN '130'. DELETE t_mach WHERE col130 <> <fs_val>.
WHEN '131'. DELETE t_mach WHERE col131 <> <fs_val>.
WHEN '132'. DELETE t_mach WHERE col132 <> <fs_val>.
WHEN '133'. DELETE t_mach WHERE col133 <> <fs_val>.
WHEN '134'. DELETE t_mach WHERE col134 <> <fs_val>.
WHEN '135'. DELETE t_mach WHERE col135 <> <fs_val>.

```

        WHEN '136'. DELETE t_mach WHERE col136 <> <fs_val>.
        WHEN '137'. DELETE t_mach WHERE col137 <> <fs_val>.
        WHEN '138'. DELETE t_mach WHERE col138 <> <fs_val>.
        WHEN '139'. DELETE t_mach WHERE col139 <> <fs_val>.
        WHEN '140'. DELETE t_mach WHERE col140 <> <fs_val>.

```

```

    ENDCASE.

```

```

ENDIF.

```

```

*****ELIMINAR RESTRICCIONES SEGUN COLUMNA*****

```

```

    IF colum_rest >= wa_configu_2-columna.
        EXIT.
    ENDIF.
ENDLOOP.

```

```

*          DO wa_configu_2-columna TIMES.
*          ADD 1 TO colum_rest.
*
*          ENDDO.

```

```

IF wa_configu_2-columna_val = w_config-columna.
    LOOP AT t_mach .
        CONCATENATE 't_mach-col' wa_configu_2-columna INTO nombre .
        ASSIGN (nombre) TO <fs_val>.

```

```

        w_valor-valor = <fs_val>.
        w_valor-fila = t_mach-fila.
        APPEND w_valor TO t_valor.
        CLEAR w_valor.

```

```

    ENDLOOP.

```

```

    lv_restriccion = 'X'.

```

```

ELSE.

```

```

        LOOP AT t_vc WHERE titulo = w_config-
titulo AND subtitulo = w_config-subtit AND caracterica = w_config-
caract AND subcaracteristica = w_config-subcar.
            w_valor-valor = t_vc-valor.

```

```

        w_valor-fila =      t_vc-ord_tit.
        APPEND w_valor TO t_valor.
        CLEAR w_valor.
    ENDLOOP.
ENDIF.

ELSE.
    LOOP AT t_vc WHERE titulo = w_config-
titulo AND subtitulo = w_config-subtit AND caracterica = w_config-
caract AND subcaracteristica = w_config-subcar.
        w_valor-valor =      t_vc-valor.
        w_valor-fila =      t_vc-ord_tit.
        APPEND w_valor TO t_valor.
        CLEAR w_valor.
    ENDLOOP.

ENDIF.

IF t_mach[] IS NOT INITIAL.
    READ TABLE t_mach INTO wa_mach INDEX 1.
ENDIF.

*****
*
SORT t_valor  ASCENDING BY valor fila ASCENDING.
DELETE ADJACENT DUPLICATES FROM t_valor COMPARING valor.

READ TABLE gt_cotiza_outtab INDEX p_es_row_no-row_id
ASSIGNING <fv_cotiza>.
* *****

CASE p_e_fieldname.
    WHEN 'VALOR'.

ENDCASE.

DATA: dmpv_nom_campo TYPE string.
DATA: dmpv_nom_config TYPE string.
FIELD-SYMBOLS: <fs> TYPE any.

IF w_config-columna < 80 .

```

```

lv_nom_campo = 'wa_plantilla1-col'.
lv_nom_config = 'WA_FLAG1-COL'.
ELSE.
  lv_nom_campo = 'wa_plantilla2-col'.
  lv_nom_config = 'WA_FLAG1-COL'.
ENDIF.
CONCATENATE lv_nom_campo w_config-columna INTO lv_nom_campo.
CONCATENATE lv_nom_config w_config-columna INTO lv_nom_config.
CONDENSE lv_nom_campo.
CONDENSE lv_nom_config.
ASSIGN (lv_nom_campo) TO <fs>.
ASSIGN (lv_nom_config) TO <config>.
READ TABLE t_valor INTO w_valor WITH KEY valor = p_e_value.
IF sy-subrc <> 0.
  <config> = ''.
ELSE.
  IF lv_restriccion = 'X'.
    READ TABLE t_valor INTO w_valor WITH KEY valor = p_e_value.
    READ TABLE t_mach INTO wa_mach WITH KEY fila = w_valor-fila.
    LOOP AT t_mach.
      CONCATENATE 'wa_mach-col' wa_configu_2-columna INTO nombre.
      ASSIGN (nombre) TO <fs_val2>.
      CONCATENATE 't_mach-col' wa_configu_2-columna INTO nombre.
      ASSIGN (nombre) TO <fs_val>.
      IF <fs_val> <> <fs_val2>.
        DELETE t_mach INDEX sy-tabix.
      ENDIF.
    ENDLOOP.
  ELSE.
    READ TABLE t_mach INTO wa_mach INDEX 1.
  ENDIF.

DATA: cont(3) TYPE n,
      colum(3) TYPE n.
cont = w_config-columna.
colum = 140 - cont.

DO colum TIMES.
  ADD 1 TO cont.
  IF cont < 80.
    CONCATENATE 'wa_plantilla1-col' cont INTO nombre.

```

```

        ASSIGN (nombre) TO <fs_val>.
ELSE.
    CONCATENATE 'wa_plantilla2-col' cont INTO nombre.
    ASSIGN (nombre) TO <fs_val>.
ENDIF.

READ TABLE lt_configu_2 INTO wa_configu_2 WITH KEY columna_val
= cont.
IF sy-subrc = 0.
    CONCATENATE 'wa_mach-col' wa_configu_2-columna INTO nombre.
    ASSIGN (nombre) TO <fs_val2>.
    <fs_val> = <fs_val2>.
ELSE.
    READ TABLE t_config WITH KEY columna = cont.
    IF sy-subrc = 0.
        READ TABLE t_vc WITH KEY titulo = t_config-
titulo subtitulo = t_config-subtit
                                caracterica = t_config-
caract subcaracteristica = t_config-subcar.
        IF sy-subrc = 0.
            <fs_val> = t_vc-valor.
        ENDIF.
    ENDIF.
ENDIF.

ENDDO.

cont = w_config-columna - 1 .
DO colum TIMES.
    ADD 1 TO cont.
    IF cont < 80.
        CONCATENATE 'wa_flag1-col' cont INTO nombre.
        ASSIGN (nombre) TO <fs_val>.
        <fs_val> = 'X'.

    ELSE.
        CONCATENATE 'wa_flag2-col' cont INTO nombre.
        ASSIGN (nombre) TO <fs_val>.
        <fs_val> = 'X'.

    ENDIF.

```

```

        ENDDO.
    ENDIF.

    <fs> = p_e_value.

*
* *****
*
*     <fv_cotiza>-valor = <fv_cotiza>-valor.
*     <fv_cotiza>-icon = '@5B@'.
*     <fv_cotiza>-st_filas = ''.

    PERFORM recalcula_precios.

    PERFORM formatear_plantilla TABLES gt_cotiza_outtab.

    PERFORM procesa_data.

    DATA: is_stable TYPE lvc_s_stbl.
    is_stable-row = 'X'.
    is_stable-col = 'X'.

    grid_cotiza->refresh_table_display(
        EXPORTING
            is_stable      = is_stable      " Without Sort, Filter, etc.
        EXCEPTIONS
            finished       = 1
            OTHERS         = 2
    ).

    grid_cotiza->get_scroll_info_via_id(
IMPORTING
    es_row_no    = es_row_no
    es_row_info  = es_row_info
    es_col_info  = es_col_info
    ).

*     ENDIF.

*     ENDIF.

```

```

ENDIF.

ENDIF.
ENDFORM.

*&-----*
*&      Form  ENVIA_CORREO
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*

FORM envia_correo USING p_numcot tipo_lib.

DATA: lwa_cab TYPE ztpp_cotiza_cab.
DATA: dmpv_text(30) TYPE c.
DATA: dmpv_monto(22) TYPE c.
DATA: lt_mailltxt          TYPE STANDARD TABLE OF soli          WITH HEADER L
INE.
DATA: dmpv_texto          TYPE so_text255.
DATA: send_request TYPE REF TO cl_bcs.
DATA: str_subject(50) .
DATA: t_hex TYPE solix_tab.
DATA: document TYPE REF TO cl_document_bcs.
DATA: sender TYPE REF TO cl_cam_address_bcs.
DATA: recipient TYPE REF TO if_recipient_bcs.
DATA: bcs_exception TYPE REF TO cx_bcs.
DATA: status_mail TYPE bcs_stml.
DATA: lx_document_bcs TYPE REF TO cx_document_bcs.
DATA: vl_email_fornecl TYPE adr6-smtp_addr.
DATA: lt_lib TYPE ztpp_liberacion OCCURS 0 WITH HEADER LINE.
DATA: txt_lib TYPE string.

DATA: dmpv_address TYPE bapiaddr3.
DATA: dmpv_return TYPE STANDARD TABLE OF bapiret2.

SELECT SINGLE * FROM ztpp_cotiza_cab INTO lwa_cab WHERE num_cot = p_num
cot.
* SELECT * FROM ztpp_liberacion INTO TABLE lt_lib WHERE NUM_COT = p_numc
ot AND ESTATUS = SPACE.

```

```

IF sy-subrc = 0.
  READ TABLE lt_liberacion WITH KEY gr_libera = tipo_lib.'"EST_LIBERA_N
UEV'.
  IF sy-subrc = 0.
    txt_lib = lt_liberacion-descrip.

    CALL FUNCTION 'BAPI_USER_GET_DETAIL'
      EXPORTING
        username = lt_liberacion-usuario
*        CACHE_RESULTS = 'X'
      IMPORTING
        address = lv_address
      TABLES
        return = lv_return.

    "lv_ADDRESS-E_MAIL

*   CONCATENATE lt_lib-descrip txt_lib INTO txt_lib SEPARATED BY '/'.
  ENDF.
*   READ TABLE lt_lib WITH KEY GR_LIBERA = tipo_lib.'"EST_LIBERA_GRAL'.
*   if sy-subrc = 0.
*     CONCATENATE lt_lib-descrip txt_lib INTO txt_lib SEPARATED BY '/'.
*   endif.
*   READ TABLE lt_lib WITH KEY GR_LIBERA = tipo_lib.'"EST_LIBERA_MARG'.
*   if sy-subrc = 0.
*     CONCATENATE lt_lib-descrip txt_lib INTO txt_lib SEPARATED BY '/'.
*   endif.

  lt_maihtxt = '<html><head>'.
  APPEND lt_maihtxt. CLEAR lt_maihtxt.
  lt_maihtxt = '<body style="font-family: Arial, Helvetica, sans-
serif; font-size: 12">'.
  APPEND lt_maihtxt. CLEAR lt_maihtxt.
  lt_maihtxt = '<p>A través del presente, se solicita la aprobación del
siguiente requerimiento:</p>'.
  APPEND lt_maihtxt. CLEAR lt_maihtxt.
  CONCATENATE lwa_cab-tip_vta lwa_cab-mod_bus lwa_cab-tip_car lwa_cab-
ano_mod INTO lv_texto SEPARATED BY '-'.

```



```

CONCATENATE '<p><b>' lv_texto '</b></p>' INTO lt_mailtxt.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<table style="font-size: 13" border="1">'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<tr>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<TD><b>Tipo requerimiento: </b></TD>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
CLEAR lv_texto.
CONCATENATE '<TD>' txt_lib '</TD>' INTO lv_texto.
* lt_mailtxt = '<TD>[margen ganancia/liberación general/liberacion requere
rimiento] [VALOR REQUERIMIENTO]</TD>'.
lt_mailtxt = lv_texto.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '</tr>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<tr>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<TD><b>Cotización: </b></TD>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
CONDENSE lwa_cab-num_cot.
CONCATENATE '<TD>' lwa_cab-num_cot '</TD>' INTO lt_mailtxt.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '</tr>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.
lt_mailtxt = '<tr>'.

lt_mailtxt = '<TD><b>Tipo Cotización: </b></TD>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.

IF lwa_cab-tp_cotiza = 1.
  lv_texto = 'Licitación'.
ELSEIF lwa_cab-tp_cotiza = 2.
  lv_texto = 'Nacional'.
ELSEIF lwa_cab-tp_cotiza = 3.
  lv_texto = 'Exportación'.
ENDIF.
CONCATENATE '<TD>' lv_texto '</TD>' INTO lt_mailtxt.

lt_mailtxt = '<TD><b>Cliente: </b></TD>'.
APPEND lt_mailtxt. CLEAR lt_mailtxt.

```

```

CONDENSE lwa_cab-kunnr2_1.
CONCATENATE '<TD>' lwa_cab-kunnr2_1 '</TD>' INTO lt_mai1txt.

lt_mai1txt = '<TD><b>Precio Total: </b></TD>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lv_monto = lwa_cab-pv_final.
CONDENSE lv_monto.
CONCATENATE '<TD>' lv_monto lwa_cab-
moneda '</TD>' INTO lt_mai1txt SEPARATED BY space.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '</tr>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '</table>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '</br>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '<p style="font-
size: 11"><b>Si necesita realizar una verificación adicional, ingresar a
la transacción SAP "ZCOTIZACION"</b></p>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '<p style="font-size: 11">Cordialmente, </p> '.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '<p><b>MODASA</b></p>'.
APPEND lt_mai1txt. CLEAR lt_mai1txt.
lt_mai1txt = '<p style="font-
size: 12">"Por favor no responda sobre este correo electrónico, es enviad
o de forma automática"</p> '.

vl_email_fornecl = lv_address-e_mail.
TRY.
send_request = cl_bcs=>create_persistent( ).

DATA: perio TYPE string.
CONCATENATE '' '' INTO perio.

CONCATENATE 'Alerta Cotización' '' INTO str_subject SEPARATED BY
space.

```

```

*      generar cuerpo del mensaje
create_internet_address( 'alertas@modasa.com.pe' ).

*      Add sender

      recipient = cl_cam_address_bcs=>create_internet_address( vl_email
_fornecl ).

      CATCH cx_document_bcs INTO lx_document_bcs.

      CATCH cx_bcs INTO bcs_exception.

      ENDTRY.
    ENDIF.
  ENDFORM.
*****
***
*&-----*
*&      Form  FORMATEAR_PLANTILLA
*&-----*
*      text
*-----*
*      <--P_GT_COTIZA_OUTTAB[] text
*-----*
*&-----*
*&      Form  FORMATEAR_PLANTILLA
*&-----*
*      text
*-----*
*      -->P_GT_COTIZA_OUTTAB text
*-----*
FORM formatear_plantilla TABLES lt_cotiza_outtab STRUCTURE gt_cotiza_ou
ttab.
      "Insertar nombre correcto para <...>.
      DATA: lw_cotiza TYPE LINE OF gt_cotiza_outtab.

      DATA:      nombre_campo TYPE string,

```

```

        nombre_config TYPE string.
DATA: segmento(3).
FIELD-SYMBOLS: <fs_cotiza> TYPE LINE OF gt_cotiza_outtab,
               <valor>     TYPE any,
               <config>    TYPE any.
DATA: titulo(50),
      subtitulo(50),
      caracteristica(50),
      subcaracteristica(50),
      contador TYPE i.
REFRESH lt_cotiza_outtab.

LOOP AT t_config.

    lw_cotiza-cod_ref2      = wa_plantilla1-codigo.
    lw_cotiza-secuen_ref2 = wa_plantilla1-fila.
    lw_cotiza-visualizar = t_config-
visualizar.      "ADD DB7 VISUALIZAR CARACT EN COTIZAC
    IF t_config-columna < 80.
        CONCATENATE 'WA_PLANTILLA1-COL' t_config-columna INTO nombre_campo.
        CONCATENATE 'WA_FLAG1-COL' t_config-columna INTO nombre_config.
    ELSE.
        CONCATENATE 'WA_PLANTILLA2-COL' t_config-columna INTO nombre_campo.
        CONCATENATE 'WA_FLAG2-COL' t_config-columna INTO nombre_config.
    ENDIF.
    ASSIGN (nombre_campo) TO <valor>.
    ASSIGN (nombre_config) TO <config>.

    IF t_config-titulo = 'ESTANDAR'.
        EXIT.
    ENDIF.

    IF t_config-titulo = '/'.
        segmento = 'X'.
        CONTINUE.
    ENDIF.

    IF t_config-titulo = '/' AND segmento = 'X'.
        segmento = 'XX'.
    ENDIF.

    IF segmento = ''.

```

```

IF t_config-titulo <> titulo AND t_config-titulo IS NOT INITIAL.
  lw_cotiza-titulo      = t_config-titulo.
  ADD 1 TO contador.
  lw_cotiza-ord_tit = contador.
  lw_cotiza-visualizar = t_config-
visualizar.      "ADD DB7 VISUALIZAR CARACT EN COTIZAC
  APPEND lw_cotiza TO lt_cotiza_outtab.
  CLEAR: lw_cotiza.
  titulo = t_config-titulo.
ENDIF.

```

```

IF t_config-subtit <> subtitulo AND t_config-subtit t_config-
caract IS NOT INITIAL ..
  lw_cotiza-titulo      = t_config-titulo.
  lw_cotiza-subtitulo   = t_config-subtit.
  ADD 1 TO contador.
  lw_cotiza-ord_tit = contador.
  lw_cotiza-visualizar = t_config-
visualizar.      "ADD DB7 VISUALIZAR CARACT EN COTIZAC
  APPEND lw_cotiza TO lt_cotiza_outtab.
  CLEAR: lw_cotiza.
  titulo = t_config-titulo.
  subtitulo = t_config-subtit.
ENDIF.

```

```

IF t_config-caract <> caracteristica AND t_config-
caract IS NOT INITIAL AND t_config-subcar IS NOT INITIAL ..
  lw_cotiza-titulo      = t_config-titulo.
  lw_cotiza-subtitulo   = t_config-subtit.
  lw_cotiza-caracterica = t_config-caract.
  ADD 1 TO contador.
  lw_cotiza-ord_tit = contador.
  lw_cotiza-visualizar = t_config-
visualizar.      "ADD DB7 VISUALIZAR CARACT EN COTIZAC
  APPEND lw_cotiza TO lt_cotiza_outtab.
  CLEAR: lw_cotiza.
  titulo      = t_config-titulo.
  subtitulo   = t_config-subtit.
  caracteristica = t_config-caract.
ENDIF.

```



```

*&-----*
*      text
*-----*
*      -->P_DMP200_NUMCOT  text
*-----*
FORM obtener_nro_cotizacion USING      p_dmp200_numcot.
  DATA: dmpv_message(80) TYPE c.

  PERFORM cargar_constantes .
  PERFORM obtener_cabecera .
  PERFORM inicializar_valores.
  DATA smp_dyntxt TYPE smp_dyntxt.
  CLEAR smp_dyntxt.
  smp_dyntxt-text = 'Visualizar'.
  smp_dyntxt-icon_id = '@0Z@'.
  smp_dyntxt-icon_text = 'Modificar'.
  smp_dyntxt-quickinfo = 'Modificar'.
  sscrfields-functxt_01 = smp_dyntxt.

  dmp200_numcot = p_dmp200_numcot.
  SELECT SINGLE   cod_ref  kunnr  kunnr2  land1 zterm vendedor  tip_vta
matnr  mod_bus tip_car  cantidad
  ano_mod  ano_fab  tip_placa  longitud alto vol_del ancho dee vol_post p
v_netto flete igv dcto cost_serv cost_man_ob pv_final
  marg_ganancha_r marg_gananc_r marg_ganancha marg_gananc ruc dni nombre
_cli contac correo telefono
  direccion prec_chasis prec_carro pv_netto_r igv_r incoterm moneda t_c
amb kunnr2_l  vendedor_l  aceptado tp_cotiza f_v_cot_d f_v_cot_h
  plazo_ent secuen_ref  secuen_ref2  confirmado aprobado rechazado ano_fa
b_carr

  INTO

  (dmp200_producto, dmp200_kunnr, dmp200_kunnr2, dmp200_land1, dmp2
00_zterm, dmp200_vendedor, dmp200_tip_vta, dmp200_matnr, dmp200_mod_bu
s, dmp200_tip_car,
  dmp200_cant, dmp200_ano_mod, dmp200_ano_fab, gv_tip_placa, dmp200
_long, dmp200_alto, dmp200_vol_del, dmp200_ancho, dmp200_dee, dmp200
_vol_post,

```

```

    dmp200_prec_netto_f, dmp200_flete_f, dmp200_igv_f, dmp200_dcto_f, g
v_cost_serv, gv_cost_man_ob, dmp200_pv_total, dmp200_margen, p_marg,
dmp200_particip_chasis_f, dmp200_particip_carroc_f,
    dmp300_c_ruc, dmp300_c_dni, dmp300_c_nombre, dmp300_c_contacto, d
mp300_c_correo, dmp300_c_telefono, dmp300_c_direccion, dmp200_prec_cha
sis,
    dmp200_prec_carroc, dmp200_prec_netto, dmp200_igv,dmp200_incoterm, l
_m1, p_t_camb,dmp200_l_kunnr2,dmp200_l_vendedor,dmp200_aceptado,dmp200_ti
p_cot,dmp200_f_v_cot_d,dmp200_f_v_cot_h,dmp200_plazo_ent,dmp200_m_text_s,
dmp200_m_text,
    dmp200_confirmado, dmp200_aprobado, dmp200_rechazado, dmp200_ano_fab_
carr )

```

```

FROM ztpp_cotiza_cab

```

```

    WHERE num_cot = p_dmp200_numcot.

```

```

IF l_m1 = 'USD'.

```

```

    c_usd = 'X'.

```

```

ELSEIF l_m1 = 'PEN'.

```

```

    c_pen = 'X'.

```

```

ENDIF.

```

```

l_m2 = l_m1.

```

```

l_m3 = l_m1.

```

```

l_m4 = l_m1.

```

```

l_m5 = l_m1.

```

```

l_m6 = l_m1.

```

```

p_descu = dmp200_dcto_f.

```

```

p_flete = dmp200_flete_f.

```

```

IF sy-subrc NE 0 AND ( ge_ucomm = 'V' OR ge_ucomm = 'M' ).

```

```

    CONCATENATE 'No Existe Cotizacion' p_dmp200_numcot INTO lv_message.

```

```

    CLEAR mp200_estado_cotiza.

```

```

    MESSAGE lv_message TYPE 'S' DISPLAY TYPE 'E'.

```

```

ELSE.

```

```

    IF dmp200_aceptado = 'X'.

```

```

        mp200_estado_cotiza = 'Liberado'.

```

```

    ELSE.

```

```

        mp200_estado_cotiza = 'Bloqueado'.

```

```

    ENDIF.

```



```

IF dmp200_confirmado = 'X'.
    mp200_estado_cotiza = 'Confirmado por Cliente'.
ENDIF.

IF dmp200_aprobado = 'X'.
    mp200_estado_cotiza = 'Aprobado por Comercial'.
ENDIF.

IF dmp200_rechazado = 'X'.
    mp200_estado_cotiza = 'Rechazado'.
ENDIF.

dmp300_c_kunnr = dmp200_kunnr.
*      CLEAR: DMP300_C_KUNNR.
ENDIF.
*      REFRESH gt_cotiza_outtab.
*DMP200_PRODUCO

DATA: lwa_cotiza_det1 TYPE ztpp_cotiza_det1.
DATA: lwa_cotiza_det2 TYPE ztpp_cotiza_det2.
DATA: lwa_cotiza_st1 TYPE ztpp_cotiza_st1.
DATA: lwa_cotiza_st2 TYPE ztpp_cotiza_st2.
DATA: lwa_cotiza_conf TYPE STANDARD TABLE OF ztpp_cotiza_conf WITH HEAD
ER LINE.

SELECT SINGLE * FROM ztpp_cotiza_det1 INTO lwa_cotiza_det1 WHERE num_c
ot = p_dmp200_numcot.
SELECT SINGLE * FROM ztpp_cotiza_det2 INTO lwa_cotiza_det2 WHERE num_c
ot = p_dmp200_numcot.
SELECT SINGLE * FROM ztpp_cotiza_st1 INTO lwa_cotiza_st1 WHERE num_cot
= p_dmp200_numcot.
SELECT SINGLE * FROM ztpp_cotiza_st2 INTO lwa_cotiza_st2 WHERE num_cot
= p_dmp200_numcot.
SELECT * FROM ztpp_cotiza_conf INTO TABLE lwa_cotiza_conf WHERE num_cot
= p_dmp200_numcot.

REFRESH t_config.
CLEAR: wa_plantilla1,wa_plantilla2,wa_flag1,wa_flag2.
LOOP AT lwa_cotiza_conf .
    CLEAR t_config.

```

```

MOVE-CORRESPONDING lwa_cotiza_conf TO t_config.
*      t_config-num_cot = dmp200_numcot.
APPEND t_config.
ENDLOOP.
MOVE-CORRESPONDING lwa_cotiza_det1 TO wa_plantilla1.
MOVE-CORRESPONDING lwa_cotiza_det2 TO wa_plantilla2.
MOVE-CORRESPONDING lwa_cotiza_st1 TO wa_flag1.
MOVE-CORRESPONDING lwa_cotiza_st2 TO wa_flag2.
PERFORM formatear_plantilla TABLES gt_cotiza_outtab.

CLEAR: c_usd, c_pen.

IF l_m1 = 'USD'.
  c_usd = 'X'.
ELSEIF l_m1 = 'PEN'.
  c_pen = 'X'.
ENDIF.

PERFORM obtner_plantilla_visual USING dmp200_producto.
CALL SCREEN 200.
ENDFORM.
*&-----*
*&      Form CONFIRMAR_COTIZACION
*&-----*

FORM confirmar_cotizacion .
DATA: dmpv_xcpdk TYPE knal-xcpdk.
DATA: dmpv_kunnr TYPE knal-kunnr.
lv_kunnr = dmp200_kunnr.
IF gv_ucomm2 = 'X'.

  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
*      TITLEBAR           = ' '
*      DIAGNOSE_OBJECT    = ' '
      text_question       = 'DESEA CONFIRMAR LA COTIZACIÓN?' .

IF answer = 1.
  IF mp200_estado_cotiza = 'Liberado'.
    mp200_estado_cotiza = 'Confirmado por Cliente'.

```

```

dmp200_confirmado = 'X'.

UPDATE ztpp_cotiza_cab SET confirmado = dmp200_confirmado
WHERE num_cot = dmp200_numcot.
MESSAGE 'Se confirmo la cotizacion' TYPE 'S'.          IF sy-
subrc = 0 AND lv_xcpdk = 'X'.

PERFORM mail_crea_cliente.
*   ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDFORM.
FORM rechazar_cot .
DATA: answer .

FIELD-SYMBOLS: <fa_rechazo> TYPE LINE OF t_rechazo.
CALL FUNCTION 'POPUP_TO_CONFIRM'
EXPORTING
*   TITLEBAR           = ' '
*   DIAGNOSE_OBJECT    = ' '
text_question          = 'SEGURO DE RECHAZAR COTIZACION?'.
IF answer = 1.

CLEAR: dmp200_motivo_rech,
       dmp200_obs_rech.
CALL SCREEN 700 AT 105 10.

IF gv_flag_rech = 'X'.
READ TABLE t_rechazo ASSIGNING <fa_rechazo> WITH KEY secuencia
= dmp200_mot_rech.
IF sy-subrc = 0.
dmp200_motivo_rech = <fa_rechazo>-valor1.
ENDIF.

```

```

UPDATE   ztpp_cotiza_cab SET   motivo_rech = dmp200_motivo_rech
                                observ      = dmp200_obs_rech
                                rechazado = 'X'
                                WHERE num_cot = dmp200_numcot.

mp200_estado_cotiza = 'Rechazado'.
dmp200_rechazado = 'X'.
ge_ucomm = 'V'.
MESSAGE 'Se rechazo la Cotizacion' TYPE 'S' DISPLAY TYPE 'E'.
ENDIF.
ENDIF.

ENDFORM.
FORM mail_crea_cliente .
    PERFORM obtener_correo_trabajadores.
ENDFORM.

FORM obtener_correo_trabajadores.

TYPES: BEGIN OF szadr_so_key_line2,
        so_key      TYPE adcp-so_key,
        addrnumber TYPE adcp-addrnumber,
        persnumber  TYPE adcp-persnumber,
        addr_type   TYPE szad_field-addr_type,
        error_flag  TYPE szad_field-flag,
    END OF szadr_so_key_line2.

DATA: v_objtp TYPE sood-objtp,
      v_objyr TYPE sood-objyr,
      v_objno TYPE sood-objno.
DATA: l_dli_id TYPE soodk,
      g_owner  TYPE soud-usrnam.

DATA: member  TYPE STANDARD TABLE OF sodm1 WITH HEADER LINE,
      objpara TYPE STANDARD TABLE OF selc,
      objparb TYPE STANDARD TABLE OF soop1.

DATA: adrnr TYPE soud-usradr.

DATA: szadr_so_key_tab TYPE STANDARD TABLE OF szadr_so_key_line2 WITH H

```

EADER LINE.

```
REFRESH: gt_correo.  
DATA: user TYPE soud3.  
SELECT SINGLE objjtp objjyr objjno INTO (v_objjtp, v_objjyr, v_objjno) FROM s  
ood
```

```
WHERE objnam = 'ZSD_COTIZ_01'.
```

```
l_dli_id-objjtp = v_objjtp.
```

```
l_dli_id-objjyr = v_objjyr.
```

```
l_dli_id-objjno = v_objjno.
```

```
CONCATENATE v_objjtp v_objjyr v_objjno INTO l_dli_id.
```

```
g_owner = sy-uname.
```

```
LOOP AT member .
```

```
IF member-memtp = 'USR'.
```

```
REFRESH: szadr_so_key_tab.
```

```
CLEAR : user, adrn.
```

```
user-usrtp = member-memtp.
```

```
user-usryr = member-memyr.
```

```
user-usrno = member-memno.
```

```
CALL FUNCTION 'SO_USER_ADDRESS_NUMBER_GET'
```

```
EXPORTING
```

```
user = user
```

```
deleted_included = 'X'
```

```
IMPORTING
```

```
user_address = adrn
```

```
EXCEPTIONS
```

```
office_name_not_exist = 19
```

```
parameter_error = 23
```

```
sap_name_not_exist = 29
```

```
user_not_exist = 34
```

```
x_error = 1000.
```

```
szadr_so_key_tab-so_key = adrn.
```

```
APPEND szadr_so_key_tab.
```

```
CALL FUNCTION 'ADDR_SO_KEY_GET_KEY'
```

```
TABLES
```

```
key_table = szadr_so_key_tab
```

```
EXCEPTIONS
```

```
internal_error = 1
```

```
no_entries_in_key_table = 2
```

```

        OTHERS = 3.
    READ TABLE szadr_so_key_tab INDEX 1.
    SELECT smtp_addr INTO TABLE gt_correo FROM adr6
    WHERE persnumber = szadr_so_key_tab-persnumber.
*     wa_correo-correo = 'csarazu@iterpconsulting.com'.
    LOOP AT gt_correo INTO wa_correo.
*     PERFORM ENVIA_CORREO USING WA_CORREO-CORREO.
    ENDLOOP.

    ELSEIF member-memtp = 'ADR'.
        DATA: correo1 TYPE adr6-smtp_addr.
        wa_correo-correo = member-address.
        APPEND wa_correo TO gt_correo.

*     PERFORM ENVIA_CORREO USING CORREO1.

    ENDIF.
    ENDLOOP.

    LOOP AT gt_correo INTO wa_correo.
        PERFORM alerta_cpd USING wa_correo-correo.
    ENDLOOP.

*GT_CORREO

    ENDFORM.                                "obtener_correo_trabajadores

    FORM alerta_cpd USING correo.
        DATA: lwa_cab TYPE ztpp_cotiza_cab.
        DATA: dmpv_text(30) TYPE c.
        DATA: dmpv_monto(22) TYPE c.
        DATA: lt_mailtxt          TYPE STANDARD TABLE OF soli      WITH HEADER L
    INE.
        DATA: dmpv_texto          TYPE so_text255.
        DATA: send_request TYPE REF TO cl_bcs.
        DATA: str_subject(50) .
        DATA: t_hex TYPE solix_tab.
        DATA: document TYPE REF TO cl_document_bcs.
        DATA: sender TYPE REF TO cl_cam_address_bcs.
        DATA: recipient TYPE REF TO if_recipient_bcs.
        DATA: bcs_exception TYPE REF TO cx_bcs.

```

```

DATA: status_mail TYPE bcs_stml.
DATA: lx_document_bcs TYPE REF TO cx_document_bcs.
DATA: vl_email_fornecl TYPE adr6-smtp_addr.
DATA: dmpv_address TYPE bapiaddr3.
DATA: dmpv_return TYPE STANDARD TABLE OF bapiret2.

lv_address = correo.
lt_maihtxt = '<html><head>'.
APPEND lt_maihtxt. CLEAR lt_maihtxt.
lt_maihtxt = '<body style="font-family: Arial, Helvetica, sans-
serif; font-size: 12">'.
APPEND lt_maihtxt. CLEAR lt_maihtxt.
lt_maihtxt = '<p>A través del presente, se requiere generar el siguien
e código de dato maestro cliente SAP:</p>'.

CONCATENATE '<TD>' dmp200_numcot '</TD>' INTO lv_texto.
lt_maihtxt = lv_texto.

* CONDENSE lwa_cab-num_cot.
CONCATENATE '<TD>' dmp300_c_dni '</TD>' INTO lt_maihtxt.

CONCATENATE '<TD>' dmp300_c_nombre '</TD>' INTO lt_maihtxt.

CONCATENATE '<TD>' dmp300_c_contacto '</TD>' INTO lt_maihtxt.

CONCATENATE '<TD>' dmp300_c_correo '</TD>' INTO lt_maihtxt.

CONCATENATE '<TD>' lv_monto '</TD>' INTO lt_maihtxt SEPARATED BY space
.

lt_maihtxt = '<TD><b>Dirección: </b></TD>'.
APPEND lt_maihtxt. CLEAR lt_maihtxt.
CONCATENATE '<TD>' dmp300_c_direccion '</TD>' INTO lt_maihtxt SEPARATE
D BY space.

lt_maihtxt = '<p style="font-
size: 11"><b>Ingresar a la transacción SAP VD01 para crear el maestro de

```

clientes con los datos mostrados.</p>'.</p></div>
<div data-bbox="138 99 886 153" data-label="Text">
<p>lt_mailtxt = '<p style="font-size: 12">"Por favor no responda sobre este correo electrónico, es enviado de forma automática"</p> '.</p></div>
<div data-bbox="138 239 354 254" data-label="Text">
<p>*lv_address = correo.</p></div>
<div data-bbox="157 260 424 275" data-label="Text">
<p>vl_email_fornecl = correo.</p></div>
<div data-bbox="157 279 204 293" data-label="Text">
<p>TRY.</p></div>
<div data-bbox="198 299 647 315" data-label="Text">
<p>send_request = cl_bcs=>create_persistent().</p></div>
<div data-bbox="198 340 444 355" data-label="Text">
<p>DATA: perio TYPE string.</p></div>
<div data-bbox="198 360 506 375" data-label="Text">
<p>CONCATENATE '' '' INTO perio.</p></div>
<div data-bbox="138 400 886 434" data-label="Text">
<p>CONCATENATE 'Alerta Cotización' '' INTO str_subject SEPARATED BY space.</p></div>
<div data-bbox="138 520 886 556" data-label="Text">
<p>sender = cl_cam_address_bcs=>create_internet_address('alertas@modasa.com.pe').</p></div>
<div data-bbox="138 642 886 676" data-label="Text">
<p>recipient = cl_cam_address_bcs=>create_internet_address(vl_email_fornecl).</p></div>
<div data-bbox="198 743 323 757" data-label="Text">
<p>COMMIT WORK.</p></div>
<div data-bbox="178 783 616 799" data-label="Text">
<p>CATCH cx_document_bcs INTO lx_document_bcs.</p></div>
<div data-bbox="178 822 505 839" data-label="Text">
<p>CATCH cx_bcs INTO bcs_exception.</p></div>
<div data-bbox="157 884 233 898" data-label="Text">
<p>ENDTRY.</p></div>
<div data-bbox="491 912 528 929" data-label="Page-Footer">
<p>205</p></div>


```

ENDFORM.
FORM autorizacion .

  IF sy-subrc = 0.

    CLEAR gv_aut_precios .
  ELSE.

    IF sy-subrc = 0.
      gv_aut_precios = 'X'.
    ELSE.
      ge_ucomm = 'V'.
    ENDIF.
  ENDIF.

  IF sy-subrc = 0.
    gv_acceso_all = 'V'.
  ENDIF.

ENDFORM.

FORM lock_table USING p_numcot.
  CALL FUNCTION 'ZENQUEUE_ZTPP_COTIZA'
    EXPORTING
      mandt      = sy-mandt
      num_cot    = p_numcot.
ENDFORM.

FORM unlock_table USING p_numcot.
  CALL FUNCTION 'ZDEQUEUE_ZTPP_COTIZA'
    EXPORTING
      mandt      = sy-mandt
      num_cot    = p_numcot
  EXCEPTIONS
    OTHERS      = 1.
ENDFORM.

```

```

FORM save_log .
DATA: st_log_cotiza      TYPE ztpp_cotiza_log,
      it_log_cotiza_cab TYPE STANDARD TABLE OF ztpp_cotiza_cab,
      st_log_cotiza_cab TYPE LINE OF it_log_cotiza_cab,
      lv_datum          TYPE sy-datum,
      lv_uzeit         TYPE sy-uzeit.

SELECT SINGLE *
      INTO st_log_cotiza_cab
      FROM ztpp_cotiza_cab
      WHERE num_cot = dmp200_numcot.
IF sy-subrc EQ 0.

lv_datum = sy-datum.
lv_uzeit = sy-uzeit.

IF st_log_cotiza_cab-f_v_cot_d NE dmp200_f_v_cot_d.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Fecha vigente del'.
  st_log_cotiza-value_old = st_log_cotiza_cab-f_v_cot_d.
  st_log_cotiza-value_new = dmp200_f_v_cot_d.
  st_log_cotiza-uname    = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-f_v_cot_h NE dmp200_f_v_cot_h.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Fecha vigente al'.
  st_log_cotiza-value_old = st_log_cotiza_cab-f_v_cot_h.
  st_log_cotiza-value_new = dmp200_f_v_cot_h.
  st_log_cotiza-uname    = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

```

```

IF st_log_cotiza_cab-tp_cotiza NE dmp200_tip_cot.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Tipo de cotización'.
  st_log_cotiza-value_old = st_log_cotiza_cab-tp_cotiza.
  st_log_cotiza-value_new = dmp200_tip_cot.
  st_log_cotiza-uname     = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

```

```

IF st_log_cotiza_cab-kunnr NE dmp200_kunnr.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Solicitante'.
  st_log_cotiza-value_old = st_log_cotiza_cab-kunnr.
  st_log_cotiza-value_new = dmp200_kunnr.
  st_log_cotiza-uname     = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

```

```

IF st_log_cotiza_cab-kunnr2 NE dmp200_kunnr2.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Cliente final'.
  st_log_cotiza-value_old = st_log_cotiza_cab-kunnr2.
  st_log_cotiza-value_new = dmp200_kunnr2.
  st_log_cotiza-uname     = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

```

```

IF st_log_cotiza_cab-zterm NE dmp200_zterm.
  CLEAR st_log_cotiza.

```

```

st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
st_log_cotiza-datum     = lv_datum.
st_log_cotiza-uzeit     = lv_uzeit.
st_log_cotiza-fieldname = 'Condición de pago'.
st_log_cotiza-value_old = st_log_cotiza_cab-zterm.
st_log_cotiza-value_new = dmp200_zterm.
st_log_cotiza-uname     = sy-uname.

MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-cantidad NE dmp200_cant.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Cantidad'.
  st_log_cotiza-value_old = st_log_cotiza_cab-cantidad.
  st_log_cotiza-value_new = dmp200_cant.
  st_log_cotiza-uname     = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-ano_fab NE dmp200_ano_fab.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Año de fabricación chasis'.
  st_log_cotiza-value_old = st_log_cotiza_cab-ano_fab.
  st_log_cotiza-value_new = dmp200_ano_fab.
  st_log_cotiza-uname     = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-ano_fab_carr NE dmp200_ano_fab_carr.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot    = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.

```

```

st_log_cotiza-fieldname = 'Año de fabricación carrocería'.
st_log_cotiza-value_old = st_log_cotiza_cab-ano_fab_carr.
st_log_cotiza-value_new = dmp200_ano_fab_carr.
st_log_cotiza-uname     = sy-uname.

MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-tip_placa NE dmp200_tip_placa.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Tipo de placa'.
  st_log_cotiza-value_old = st_log_cotiza_cab-tip_placa.
  st_log_cotiza-value_new = dmp200_tip_placa.
  st_log_cotiza-uname    = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-plazo_ent NE dmp200_plazo_ent.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Plazo de entrega'.
  st_log_cotiza-value_old = st_log_cotiza_cab-plazo_ent.
  st_log_cotiza-value_new = dmp200_plazo_ent.
  st_log_cotiza-uname    = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-flete NE dmp200_flete_f.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Flete'.
  st_log_cotiza-value_old = st_log_cotiza_cab-flete.
  st_log_cotiza-value_new = dmp200_flete_f.

```

```

st_log_cotiza-uname      = sy-uname.

MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-dcto NE dmp200_dcto_f.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot  = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Descuento'.
  st_log_cotiza-value_old = st_log_cotiza_cab-dcto.
  st_log_cotiza-value_new = dmp200_dcto_f.
  st_log_cotiza-uname    = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-marg_ganancha_r NE dmp200_margen.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot  = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Margen de chasis'.
  st_log_cotiza-value_old = st_log_cotiza_cab-marg_ganancha_r.
  st_log_cotiza-value_new = dmp200_margen.
  st_log_cotiza-uname    = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.
ENDIF.

IF st_log_cotiza_cab-marg_gananc_r NE p_marg.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot  = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum    = lv_datum.
  st_log_cotiza-uzeit    = lv_uzeit.
  st_log_cotiza-fieldname = 'Margen de carroceria'.
  st_log_cotiza-value_old = st_log_cotiza_cab-marg_gananc_r.
  st_log_cotiza-value_new = p_marg.
  st_log_cotiza-uname    = sy-uname.

  MODIFY  ztpp_cotiza_log  FROM st_log_cotiza.

```

```

ENDIF.

IF st_log_cotiza_cab-t_camb NE p_t_camb.
  CLEAR st_log_cotiza.
  st_log_cotiza-num_cot   = st_log_cotiza_cab-num_cot.
  st_log_cotiza-datum     = lv_datum.
  st_log_cotiza-uzeit     = lv_uzeit.
  st_log_cotiza-fieldname = 'Tipo de cambio'.
  st_log_cotiza-value_old = st_log_cotiza_cab-t_camb.
  st_log_cotiza-value_new = p_t_camb.
  st_log_cotiza-uname     = sy-uname.

  MODIFY ztpp_cotiza_log FROM st_log_cotiza.
ENDIF.

ENDIF.

ENDFORM.

FORM mostrar_log .
  CALL SCREEN 0201.
ENDFORM.

FORM cargar_log .
  REFRESH: it_log.

  SELECT *
    INTO TABLE it_log
    FROM ztpp_cotiza_log
    WHERE num_cot = dmp200_numcot.
ENDFORM.

*-----*
FORM crear_catalogo_dmp201 .

ENDFORM.

FORM crear_alv_dmp201.

IF dmp201_alv1 IS INITIAL.

```

```

CREATE OBJECT dmp201_cont
EXPORTING
    container_name = 'ALV_DMP201'.

CREATE OBJECT dmp201_alv1
EXPORTING
    i_parent      = dmp201_cont
    i_appl_events = 'X'.

dmp201_layo-zebra      = 'X'.
dmp201_layo-sel_mode  = 'A'.
->handle_hotspot_click for dmp201_alv1.

dmp201_vari-report = sy-cprog.

*   CREATE OBJECT go_event01.
*   SET HANDLER   go_event01->hotspot_click      FOR dmp100_alv1.

CALL METHOD dmp201_alv1->set_table_for_first_display
EXPORTING
    i_save          = 'A'
    i_default       = 'A'
    it_toolbar_excluding = dmp201_excl
    is_layout       = dmp201_layo
    is_variant      = dmp201_vari
CHANGING
*   it_sort         = t_sort
    it_outtab       = it_log
    it_fieldcatalog = dmp201_fcat[].

ELSE..

    dmp201_alv1.
ENDIF.

ENDFORM.

FORM init_tipcambio .

```



```

c_usd = 'X'.
CLEAR: c_pen.
IF c_pen = 'X'.
*   CLEAR p_t_camb.
    DATA: ls_tipo_cambio TYPE bap1093_0.
*   "si no ingresan tipo de cambio obtener el tipo de cambio de la fecha
clave
    IF p_t_camb IS INITIAL OR p_t_camb = 1.

        p_t_camb = ls_tipo_cambio-exch_rate_v.
    ENDIF.
ENDIF.
ENDFORM.

```

```

REFRESH t_cfg1.
REFRESH t_cfg2.
CLEAR status.

```

```

LOOP AT t_config.
    IF t_config-titulo = '/' AND t_config-subtit = '/' AND t_config-
caract = '/' AND t_config-subcar = '/'.
        IF status <> 'X'.
            status = 'X'.
        ELSE.
            status = 'XX'.
        EXIT.
    ENDIF.
ELSE.
    IF status = ''.
        MOVE-CORRESPONDING t_config TO t_cfg1.
        APPEND t_cfg1.
    ELSEIF status = 'X'.
        MOVE-CORRESPONDING t_config TO t_cfg2.
        APPEND t_cfg2.
    ENDIF.
ENDIF.
ENDLOOP.

```

```

LOOP AT t_cfg2.

```

```

IF t_cfg2-columna < 80.
    CONCATENATE 'wa_plantilla1-col' t_cfg2-columna INTO nombre.
ELSE.
    CONCATENATE 'wa_plantilla2-col' t_cfg2-columna INTO nombre.
ENDIF.
ASSIGN (nombre) TO <fs_val>.
CLEAR <fs_val>.
ENDLOOP.

LOOP AT t_cfg1.
    IF t_cfg1-columna < 80.
        CONCATENATE 'wa_plantilla1-col' t_cfg1-columna INTO nombre.
    ELSE.
        CONCATENATE 'wa_plantilla2-col' t_cfg1-columna INTO nombre.
    ENDIF.
    ASSIGN (nombre) TO <fs_val>.
    "OBTIENE VALOR
    LOOP AT t_vc WHERE titulo = t_cfg1-titulo AND subtitulo = t_cfg1-
subtit AND
                                caracterica = t_cfg1-
caract AND subcaracteristica = t_cfg1-subcar AND
                                valor = <fs_val>.
    READ TABLE t_cfg2 WITH KEY titulo = t_cfg1-titulo.
    IF sy-subrc = 0.
        IF t_cfg2-columna < 80.
            CONCATENATE 'wa_plantilla1-col' t_cfg2-columna INTO nombre.
        ELSE.
            CONCATENATE 'wa_plantilla2-col' t_cfg2-columna INTO nombre.
        ENDIF.
        ASSIGN (nombre) TO <fs_val2>.
        monto = <fs_val2>.
        IF t_vc-cantidad IS NOT INITIAL.
            precio_x_cantidad = t_vc-costo * t_vc-cantidad.
        ELSE.
            precio_x_cantidad = t_vc-costo.
        ENDIF.
        ADD precio_x_cantidad TO monto.
        monto_c = monto.
        <fs_val2> = monto_c.
    ENDIF.
ENDLOOP. ENDLOOP.

```

```

LOOP AT t_vc2.
*   READ TABLE lt_pr_tit ASSIGNING <fs_pr_tit> WITH KEY titulo = t_vc2-
titulo.
*   if sy-subrc = 0.
*       ADD t_vc2-costo TO <fs_pr_tit>-precio.
*   endif.
READ TABLE t_cfg2 WITH KEY titulo = t_vc2-titulo.
IF sy-subrc = 0.
    IF t_cfg2-columna < 80.
        CONCATENATE 'wa_plantilla1-col' t_cfg2-columna INTO nombre.
    ELSE.
        CONCATENATE 'wa_plantilla2-col' t_cfg2-columna INTO nombre.
    ENDIF.
    ASSIGN (nombre) TO <fs_val2>.
    monto = <fs_val2>.
    IF t_vc2-cantidad IS NOT INITIAL.
        precio_x_cantidad = t_vc2-costo * t_vc2-cantidad.
    ELSE.
        precio_x_cantidad = t_vc2-costo.
    ENDIF.

    ADD precio_x_cantidad TO monto.
    monto_c = monto.
    <fs_val2> = monto_c.
ENDIF.
ENDLOOP.

```

ENDFORM.

```

*&-----*
*&      Form  COTIZADOR
*&-----*
*      text
*-----*
*      -->P_DMP200_NUMCOT  text
*-----*

```

```

FORM cotizador USING num_cot.
  DATA: dmpv_ncot TYPE ze_pp_num_cot.
  DATA: dmpv_version2 TYPE c.
  lv_ncot = num_cot.
  FREE MEMORY ID 'ZPPP_COTIZACION_NUEVO'.
  EXPORT lv_ncot TO MEMORY ID 'ZPPP_COTIZACION_NUEVO'.
  SELECT SINGLE version2 FROM ztpp_cotiza_cab INTO lv_version2 WHERE num
_cot = lv_ncot.
  IF sy-subrc = 0.
    IF lv_version2 = 'X'.
      * SUBMIT zppp_cotizacion_final AND RETURN.
      * LEAVE PROGRAM.
    ELSE.
      SUBMIT zppp_cotizacion_nuevo AND RETURN.
      LEAVE PROGRAM.
    ENDIF.
  ENDIF.
ENDIF.
ENDFORM.

```

```

*&-----*
*&      Form  VALIDA_ARCHIVOS
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*

```

```

FORM valida_archivos .
  DATA: lt_srgbtbrel TYPE STANDARD TABLE OF srgbtbrel WITH HEADER LINE.
  DATA: lt_sood TYPE STANDARD TABLE OF sood WITH HEADER LINE.
  CLEAR gv_valida_archivos.
  CONCATENATE 'COT_' dmp200_numcot INTO objeto_carga_imagen .

  SELECT *
    INTO TABLE lt_srgbtbrel
    FROM srgbtbrel
    WHERE instid_a EQ objeto_carga_imagen AND
           reltype EQ 'ATTA'.
  IF sy-subrc EQ 0.
    SELECT *
      INTO TABLE lt_sood
      FROM sood
      FOR ALL ENTRIES IN lt_srgbtbrel

```

```
WHERE objtp EQ lt_srgbtbrel-instid_b+17(3) AND
      objyr EQ lt_srgbtbrel-instid_b+20(2) AND
      objno EQ lt_srgbtbrel-instid_b+22(12) AND
      objdes TYPE 'COT%'.
IF sy-subrc <> 0.
  gv_valida_archivos = 'X'.
ENDIF.
ELSE.
  gv_valida_archivos = 'X'.
ENDIF.
*OBTENER OBJDES CP 'NOM_ARHIVO*' DE GT_SOOD.
ENDFORM.
```

Diseño de impresión del formulario-----

28/11/2021 19:15:34

COTIZACIÓN N° 300000007
CLIENTE: 99999 - EMPRESA DE TRANSPORTES LUIS BANCHERO ROSSIS.A.
RUC : 20102234847
CLIENTE FINAL: 99999 - EMPRESA DE TRANSPORTES LUIS BANCHERO ROSSIS.A.
PRODUCTO: BUS
MATERIAL: T31UR00018
MODELO/BU S: TITAN
TIPO CARROCERÍA: URBANO
TIPO PLACA: VERDE
CANTIDAD: 10
CONTACTO: WALTER EDGAR INGARUCA TABRA
PAÍS DESTINO: PERÚ
AÑO FAB. CARROCERÍA: 2017
AÑO FAB. CHASIS: 2017
MODELO UNIDAD: MTUG



Antigua Carretera Panamericana S/r Km. 38.2
Lurin - Lima
T/ (511) 616-9800
Fax: (511) 430-0655
Av. Los frucales N° 320Ate - Lima
T/ (511) 615-8500
Fax: (511) 620-9400
www.modasa.com.pe



La imagen es una referencia del producto final.



DATOS DEL VEHICULO

DIMENSIONES	
LONGITUD:	12000 MM
ANCHO:	2560 MM
ALTURA:	3160 MM
D.E.E.	5950 MM
VOLADIZO DEL	2465 MM
VOLADIZO POST.	3585 MM

CHASIS	
*MARCA CHASIS	MERCEDES BENZ
*MODELO CHASIS	OF 1721/59
*COMBUSTIBLE	GNV
*MOTOR	CUMMINS
*MODELO MOTOR	BGE 230
*FORMULA RODANTE	4X2
*RETARDADOR DE FRENO	NO
*NUMERO DE CILINDROS GAS	9

CARROCERIA	
*CANTIDAD DE ASIENTOS	48
*ASIENTO PILOTO	GRAMMER
*ASIENTO COBRADOR	NO

SISTEMAS INTERNOS	
*FORRADO DE PISO	PLANCHA ESTRIADA 100%
*TAPASOL DE CONDUCTOR	SI

SISTEMAS EXTERNOS	
*ESPEJOS RETROVISORES	ESPEJO NUEVO

CLIMATIZACION	
*RENOVADORES DE AIRE	NO
*CLARABOYAS	3
*DE SEMPAÑADOR	AIRE CALIENTE

SISTEMA ELECTRONICO	
*RADIO CONSOLA	SI
*LETRERO DE RUTA FRONTAL	SI
*LETRERO DE RUTA LATERAL DERECHO	NO
*LETRERO DE RUTA LATERAL IZQUIERDO	NO