Módulo de Gestión de Compra-Venta de Cheques para Odoo 17 CE

Este módulo implementa la gestión completa de operaciones de compra y venta de cheques, con las siguientes características:

Características principales

Gestión completa de cheques: Registro, compra individual o múltiple, venta individual o múltiple, anulación y gestión de rechazos.

Dashboard principal: Con acceso rápido a las operaciones de compra y venta, y visualización de las últimas compras y cheques disponibles.

Flujo de estados: Borrador -> Disponible -> Vendido/Anulado/Rechazado.

Wallet de clientes/proveedores: Seguimiento de saldo en ARS para cada contacto con historial de transacciones.

Fórmulas configurables: Personalización de cálculos para pesificación, interés y precios.

Sistema de alertas avanzado:

- Alerta de vencimientos: Sistema visual de alertas para cheques próximos a vencer (30, 15, 7 días).

- Alerta de disponibilidad: Indicación de cuándo el cheque estará disponible para cobro.

Checklist de validación: Verificación de emisor, irregularidades y firma.

Gestión de fechas: Control de fecha de emisión, fecha de pago/cobro y fecha de vencimiento (calculada).

Compra múltiple: Capacidad de comprar varios cheques simultáneamente a un mismo proveedor.

Venta múltiple: Capacidad de vender varios cheques simultáneamente a un mismo cliente.

Gestión de rechazos: Proceso completo para manejar cheques rechazados, incluyendo compensaciones.

Tasas configurables: Posibilidad de modificar tasas de pesificación e interés para cada cheque o en bloque.

Adjuntos de imágenes: Posibilidad de adjuntar frente y dorso de los cheques.

Grupos de seguridad: Gestor, Supervisor y Solo lectura.

Flujo de operación

Compra de cheques:

Opción de compra individual o múltiple

Para compra múltiple: seleccionar un proveedor, agregar cheques, ajustar tasas

Completar los datos de cada cheque (fechas, montos, banco) y el checklist

Confirmar la compra (todos los cheques pasan a estado "Disponible")

Se genera un único movimiento en la wallet del proveedor con el monto total

Venta de cheques:

Opción de venta individual o múltiple

Para venta múltiple: seleccionar un cliente, agregar cheques, ajustar tasas

Confirmar la venta (todos los cheques pasan a estado "Vendido")

Se genera un movimiento en la wallet del cliente con el monto total

Gestión de rechazos:

Seleccionar un cheque en estado "Disponible" o "Vendido"

Iniciar el proceso de rechazo proporcionando un motivo

Decidir si se quiere revertir la operación de compra y/o venta

Indicar los montos de compensación (pueden diferir de los montos originales)

Se generan movimientos de compensación en la wallet del proveedor y/o cliente

Configuración

Bancos: Alta de bancos con nombre y código.

Fórmulas de cálculo: Configuración de las fórmulas para los valores calculados.

Grupos de seguridad: Asignación de permisos para las distintas operaciones.

Estructura del código

El módulo está organizado en varios archivos para facilitar el mantenimiento:

Modelos principales:

chequera\_check.py: Definición base del modelo de cheque

chequera\_check\_compute.py: Métodos de cálculo para fechas y valores

chequera\_check\_operations.py: Operaciones y acciones sobre cheques

chequera\_purchase\_wizard.py: Wizard para compra múltiple

chequera\_sale\_wizard.py: Wizard para venta múltiple

chequera\_rejection\_wizard.py: Wizard para gestión de rechazos

chequera\_wallet.py: Gestión de carteras de clientes/proveedores

chequera\_bank.py: Gestión de bancos

chequera\_formula.py: Configuración de fórmulas de cálculo

partner\_inherit.py: Extensiones al modelo de contactos

Vistas principales:

chequera\_dashboard\_view.xml: Dashboard principal

chequera\_check\_view.xml: Vistas para cheques

chequera\_purchase\_wizard\_view.xml: Vistas para el wizard de compra múltiple

chequera\_sale\_wizard\_view.xml: Vistas para el wizard de venta múltiple

chequera\_rejection\_wizard\_view.xml: Vistas para el wizard de rechazo

chequera\_wallet\_view.xml: Vistas para movimientos de wallet

chequera\_menus.xml: Estructura de menús del módulo

Dependencias

Módulo base

Módulo mail (para el tracking y chatter)

Módulo web (para assets)

Desarrollado por

VRP - Virtual Remote Partner

Información para soporte futuro

Para continuar el desarrollo en futuras sesiones, es importante tener en cuenta:

Estructura modular: El código está dividido en archivos más pequeños para facilitar el mantenimiento:

Modelo principal (chequera\_check.py)

Cálculos (chequera\_check\_compute.py)

Operaciones (chequera\_check\_operations.py)

Wizard de compra (chequera\_purchase\_wizard.py)

Wizard de venta (chequera\_sale\_wizard.py)

Wizard de rechazo (chequera\_rejection\_wizard.py)

Compatibilidad con Odoo 17 CE: Se han tenido en cuenta las particularidades de esta versión:

No se utilizan dominios con company\_type

Se han adaptado las vistas XML según las recomendaciones de Odoo 17

Se utilizan expresiones directas para atributos invisibles en lugar de attrs y states

Se respeta el orden de carga de archivos para evitar problemas de referencia

Puntos pendientes para futuras mejoras:

Mejorar la actualización de totales en tiempo real al modificar tasas

Optimizar la experiencia de usuario al agregar/editar cheques en compras/ventas múltiples

Asegurar que las tablas del dashboard siempre muestren datos actualizados

Añadir reportes estadísticos sobre cheques rechazados

Implementar un sistema de notificaciones para cheques próximos a vencer

Flujos importantes:

Los botones del dashboard inician los procesos de compra/venta directamente

La compra/venta individual son casos particulares de operaciones múltiples (1 cheque)

La actualización masiva de tasas se aplica solo a los cheques de la operación actual

Los rechazos generan movimientos de compensación registrados y rastreables

Áreas para futuros desarrollos:

Reportes y análisis de operaciones

Integración con contabilidad

Gestión avanzada de permisos y aprobaciones

Mejoras en la interfaz de usuario

Dashboard estadístico más completo con gráficos

Documentación Técnica del Módulo Chequera

Esta documentación proporciona detalles técnicos para desarrolladores que desean extender o integrar con el módulo Chequera.

Modelos disponibles para heredar/extender

chequera.check

Modelo principal que representa un cheque en el sistema.

pythonclass ChequeraCheck(models.Model):

\_name = 'chequera.check'

\_inherit = ['mail.thread', 'mail.activity.mixin']

chequera.wallet.movement

Modelo que gestiona los movimientos en las wallets de contactos.

pythonclass ChequeraWalletMovement(models.Model):

\_name = 'chequera.wallet.movement'

chequera.bank

Modelo para gestionar los bancos asociados a los cheques.

pythonclass ChequeraBank(models.Model):

\_name = 'chequera.bank'

chequera.formula\_config

Modelo para configurar fórmulas de cálculo personalizadas.

pythonclass ChequeraFormulaConfig(models.Model):

\_name = 'chequera.formula\_config'

chequera.purchase.wizard

Modelo para la compra múltiple de cheques.

pythonclass ChequeraPurchaseWizard(models.Model):

\_name = 'chequera.purchase.wizard'

chequera.sale.wizard

Modelo para la venta múltiple de cheques.

pythonclass ChequeraSaleWizard(models.Model):

\_name = 'chequera.sale.wizard'

chequera.rejection.wizard

Modelo transitorio para procesar rechazos de cheques.

pythonclass ChequeraRejectionWizard(models.TransientModel):

\_name = 'chequera.rejection.wizard'

Extensiones de modelos estándar

pythonclass ResPartner(models.Model):

\_inherit = 'res.partner'

Campos importantes que pueden ser reutilizados

En chequera.check

name: Código único del cheque (CHEQ#xxxxxx)

state: Estado del cheque (borrador, disponible, vendido, anulado, rechazado)

monto: Valor nominal del cheque

fecha\_emision: Fecha de emisión del cheque

fecha\_pago: Fecha de pago o cobro del cheque

fecha\_vencimiento: Fecha de vencimiento calculada (30 días después de fecha\_pago)

dias\_para\_vencimiento: Días restantes hasta el vencimiento

dias\_para\_disponibilidad: Días restantes hasta que el cheque esté disponible para cobro

proveedor\_id: Proveedor que vendió el cheque

cliente\_id: Cliente que compró el cheque

precio\_compra: Precio al que se compró el cheque

precio\_venta: Precio al que se vendió el cheque

alerta\_vencimiento: Estado de alerta según el vencimiento (normal, alerta\_30, alerta\_15, alerta\_7, vencido)

motivo\_rechazo: Motivo del rechazo del cheque

fecha\_rechazo: Fecha en que se registró el rechazo

En chequera.wallet.movement

partner\_id: Contacto relacionado con el movimiento

cheque\_id: Cheque relacionado (individual)

check\_ids: Cheques relacionados (múltiples)

tipo: Tipo de movimiento (compra, venta, anulacion)

monto: Importe del movimiento

multiple\_checks: Indicador de operación múltiple

es\_compensacion: Indicador de movimiento de compensación por rechazo

movement\_origin\_id: Referencia al movimiento original que se está compensando

En res.partner (extensión)

wallet\_balance: Saldo de la wallet del contacto

pesification\_rate: Tasa de pesificación predeterminada

monthly\_interest: Interés mensual predeterminado

Métodos clave que podrían ser sobrescritos o extendidos

Operaciones de compra/venta

pythondef action\_comprar(self) # Inicia proceso de compra

def action\_vender(self) # Procesa venta de un cheque

def action\_anular(self) # Anula un cheque

def action\_rechazar(self) # Rechaza un cheque

def action\_reset\_to\_draft(self) # Devuelve un cheque a estado borrador

Cálculos de valores

pythondef \_compute\_valores\_compra(self) # Calcula pesificación, interés y precio de compra

def \_compute\_valores\_venta(self) # Calcula pesificación, interés y precio de venta

def \_compute\_fecha\_vencimiento(self) # Calcula la fecha de vencimiento

def \_compute\_dias\_para\_vencimiento(self) # Calcula días para vencimiento

def \_compute\_dias\_para\_disponibilidad(self) # Calcula días para disponibilidad

def \_compute\_meses\_hasta\_vencimiento(self) # Calcula meses hasta vencimiento

Puntos de extensión de los wizards

pythondef action\_confirmar\_compra(self) # Confirma una compra múltiple

def action\_confirmar\_venta(self) # Confirma una venta múltiple

def action\_update\_tasas\_masivas(self) # Actualiza tasas masivamente

def action\_rechazar\_cheque(self) # Procesa el rechazo de un cheque

Operaciones de wallet

pythondef \_compute\_wallet\_balance(self) # Calcula el saldo de la wallet para un contacto

def action\_confirm(self) # Confirma un movimiento de wallet

def action\_cancel(self) # Anula un movimiento de wallet

Puntos de extensión previstos (hooks)

En creación de cheques

python@api.model\_create\_multi

def create(self, vals\_list)

Se pueden añadir valores predeterminados o validaciones adicionales.

En transiciones de estado

Los métodos de acción como action\_comprar, action\_vender, action\_rechazar contienen la lógica de

transición de estado y pueden ser extendidos.

En cálculos financieros

Las fórmulas en \_compute\_valores\_compra y \_compute\_valores\_venta se pueden personalizar

usando el modelo chequera.formula\_config.

En validaciones

python@api.onchange('fecha\_emision')

def \_onchange\_fecha\_emision(self)

Se pueden añadir validaciones adicionales en métodos onchange.

Convenciones de nomenclatura

Modelos

Singular: chequera.check, chequera.bank

Prefijo "chequera." para todos los modelos del módulo

Campos

Separados por guion bajo: fecha\_emision, dias\_para\_vencimiento

Sufijo \_id para campos Many2one: proveedor\_id, cliente\_id

Prefijo is\_ para campos booleanos: is\_in\_purchase\_wizard

Métodos

Públicos (para ser llamados desde fuera): action\_comprar, action\_vender

Privados (para uso interno): \_compute\_valores\_compra, \_onchange\_fecha\_emision

Decoradores para métodos computados: @api.depends, @api.onchange

Vistas

Prefijo view\_chequera\_ para vistas

Nombres descriptivos: view\_chequera\_check\_form, view\_chequera\_wallet\_tree

Eventos y señales

El módulo utiliza los siguientes eventos que pueden ser útiles para integraciones:

Creación de un nuevo cheque

Cambio de estado de un cheque (borrador → disponible → vendido/anulado/rechazado)

Creación de un movimiento en la wallet

Modificación de tasas de pesificación o interés

Rechazo de un cheque con posibles compensaciones

Consideraciones para extensiones

Al extender chequera.check, asegúrate de manejar correctamente los estados, incluyendo el estado 'rechazado'

Si añades campos relacionados con valores monetarios, considera actualizar los cálculos en \_compute\_valores\_compra y \_compute\_valores\_venta

Para integraciones contables, los momentos clave para generar asientos contables son los métodos action\_comprar, action\_vender y action\_rechazar\_cheque

La relación entre cheques y movimientos de wallet es esencial; mantén la integridad de estas relaciones en tus extensiones

Los movimientos de compensación por rechazos requieren atención especial al extender la funcionalidad