

Elaboración del módulo de impresión de factura fiscal en la facturación regular de Odoo

Versión 1.0 - noviembre 2020

Campos a generar:

Modelo: **res.config.settings**

```
printax_ip = fields.Char(string='IP del PrinTax', default='192.168.1.100', size = 16 )
printax_port = fields.Char(string='Puerto del PrinTax', default='5125', size = 6 )
```

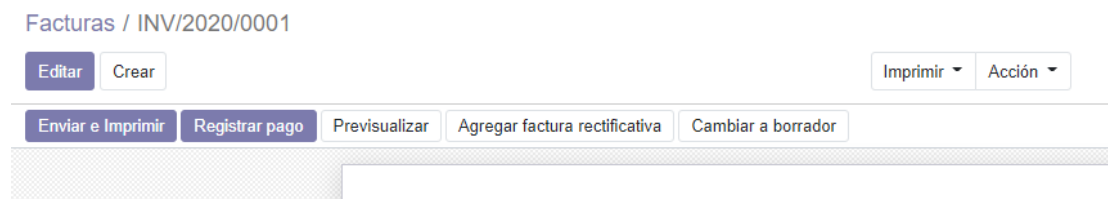
Modelo: **account.move**

```
ptx_fiscal_invoice = fields.Char("Número de factura fiscal impresa", size = 10)
ptx_serial_printer = fields.Char("Serial del impresor fiscal", size = 12)
ptx_printing_date = fields.Datetime('fecha y hora de impresión de la factura fiscal')
ptx_base_imponible = fields.Float('Total de la base imponible')
ptx_impuesto_printer = fields.Float('IVA calculado por el impresor')
ptx_reporte_z = fields.Char("Número de reporte Z asociado a la factura", size = 6)
```

En el módulo del menú: Ajustes -> Facturación

En el área "Facturas de cliente" agregar los campos printax_ip y printax_port de manera similar a la configuración del POS.

En el módulo del menú: Facturación -> Clientes -> Facturas agregar un botón "Factura fiscal".



Facturas / INV/2020/0001

Editar Crear Imprimir ▼ Acción ▼

Enviar e Imprimir Registrar pago Previsualizar Agregar factura rectificativa Cambiar a borrador

Proceso de emisión de factura fiscal.

Lo primero que se debe verificar es si ya tiene un número de factura, en caso de que lo tenga no debe imprimir nuevamente y termina el proceso. (**Próxima etapa:** Poder imprimir una copia de la factura en caso de que el usuario lo indique).

Si la factura está en borrador se debe pasar a publicado. (**Próxima etapa:** Si está impresa no se debería poder anular sin emitir una nota de crédito)

Tomando la datos del modelo **account.move**, se arma la información necesaria para enviar al impresor fiscal, los detalles del formato están en el manual de

programación del PrinTax, (OJO, es pseudocódigo, hay que colocarle los objetos y las variables correspondientes)

```
cmd = "TIP=FAC\n"
cmd += "NRF=" + str(partner.vat) + "\n"
cmd += "CNM=" + str(partner.name) + "\n"
cmd += "TXT=Número de orden " + nro_orden + "\n"
```

Se realiza el ciclo recorriendo los productos en el modelo: `account.move.line`

```
for lines in account.move.line:
    tax = self.env['account.tax'].sudo().browse(lines.tax_ids.id)
    cmd += "PRD="
    cmd += str(tax.amount) + ","
    cmd += str(account.move.line.price_unit) + ","
    cmd += str(account.move.line.qty) + ","
    cmd += account.move.line.display_name + "\n"
```

Y después de los productos se debe colocar:

```
cmd += "SUT=1\n"
cmd = base64.b64encode(cmd.encode('utf-8'))
```

Este comando se envía a la dirección y el puerto definido en **res.config.settings**.

```
respuesta = requests.get('http://' + res.config.settings.printax_ip + ':' +
    res.config.settings.printax_port + '/sendcmd/' + cmd.decode("utf-8"),
    timeout=20)
respuesta = respuesta.json()
if 'ERR' in respuesta:
```

Aquí se debe colocar un mensaje de error con el contenido de **respuesta['ERR']**

```
account.move.update({
    'ptx_fiscal_invoice': respuesta['FAC'],
    'ptx_serial_printer': respuesta['SER'],
    'ptx_printing_date': respuesta['FEC'],
    'ptx_base_imponible': respuesta['VTA'],
    'ptx_impuesto_printer': respuesta['IVA'],
    'ptx_reporte_z': respuesta['RPZ']
})
```