

MÓDULO: rrhh

1. Estructura inicial del módulo

Responsable: Backend

Tareas:

- Crear módulo Odoo base l10n_sv_hr_retenciones
- Configurar __manifest__.py con dependencias (hr, hr_payroll)
- Preparar estructura de carpetas: models/, views/, data/, security/

2. Modelos de configuración de retenciones

Responsable: Backend

Tareas:

- Crear modelo hr.retencion.iss: porcentaje, techo, aplicable a empleado/patrón
- Crear modelo hr.retencion.afp: porcentaje, techo, aplicable a empleado/patrón
- Crear modelo hr.retencion.renta: tabla escalonada con cuota fija, exceso y porcentaje
- Cargar datos de ejemplo en XML para cada tabla

3. Vista de configuración de retenciones

Responsable: Frontend + Backend

Tareas:

- Agregar nuevo menú: Configuración > Retenciones
- Crear submenús:
 - Tabla ISSS
 - Tabla AFP
 - Tabla Renta
- Crear vistas list y form para cada tabla de retención

4. Lógica de cálculo de retenciones

Responsable: Backend

Tareas:

- Heredar hr.payslip.compute_sheet
- Calcular ISSS, AFP y Renta en base al salario base del contrato
- Aplicar estos valores como input_line_ids o worked_days_line_ids según diseño
- Registrar valores negativos para que se resten automáticamente

5. Asignaciones dinámicas de salario

Responsable: Backend + Frontend

Tareas:

- Crear modelo hr.salary.assignment con:
 - Empleado
 - Tipo de asignación (hora extra, comisión, etc.)
 - Monto o cantidad
 - Fecha efectiva o período
- Relacionar con hr.employee y hr.payslip
- Crear vista (menú separado) para ingresar asignaciones

6. Menú de gestión de asignaciones

Responsable: Frontend

Tareas:

- Nuevo menú: Nómina > Asignaciones de Salario
- Crear vista list y form con filtros por período y empleado
- Permitir registrar múltiples asignaciones por empleado antes de cada ciclo de nómina

7. Reporte de nómina con detalle de retenciones y asignaciones

Responsable: Frontend

Tareas:

- Heredar plantilla QWeb del recibo (hr_payslip.report_payslip)
- Agregar detalle de:
 - Retenciones aplicadas (ISSS, AFP, Renta)
 - Asignaciones aplicadas (horas extra, etc.)
 - Totales ajustados

8. Pruebas funcionales y ajustes

Responsable: Backend + Frontend

Tareas:

- Prueba de cálculo con diferentes rangos salariales
- Verificar techo de ISSS/AFP
- Validar retención de Renta según tabla
- Probar asignaciones múltiples y combinadas
- Validar visualización correcta en la UI y reporte

9. Seguridad y accesos

Responsable: Backend

Tareas:

- Agregar reglas de acceso (ir.model.access.csv)
- Definir grupos si se desea limitar acceso a configuraciones

PROCESO DE DESARROLLO:

1. ESTRUCTURA INICIAL DEL MÓDULO

Responsable: Backend

Tareas detalladas:

- Crear carpeta del módulo: l10n_sv_hr_retenciones/
- Crear archivo __manifest__.py e incluir:
 - depends: "hr", "hr_payroll"
 - Datos de autor, versión, descripción
- Crear archivos base:
 - __init__.py
 - models/__init__.py
- Crear carpetas:
 - models/
 - views/
 - data/
 - security/
- Crear archivo security/ir.model.access.csv

2. MODELOS DE CONFIGURACIÓN DE RETENCIONES

Responsable: Backend

Tareas detalladas:

- Crear modelo hr.retencion.issv con campos:
 - porcentaje_empleado
 - porcentaje_patrono
 - techo_salarial
- Crear modelo hr.retencion.afp con campos similares
- Crear modelo hr.retencion.renta con campos:
 - salario_desde

- salario_hasta
 - cuota_fija
 - porcentaje
 - sobre_exceso_de
- Definir @api.model o @api.multi según versión
- Crear registros XML de ejemplo en data/renta_table.xml, issa_data.xml, afp_data.xml

3. VISTA DE CONFIGURACIÓN DE RETENCIONES

Responsable: Frontend + Backend

Tareas detalladas:

- Crear menú superior "Configuración" si no existe
- Agregar submenú: Configuración > Retenciones
- Crear submenús hijos:
 - Tabla ISSS → modelo hr.retencion.issa
 - Tabla AFP → modelo hr.retencion.afp
 - Tabla Renta → modelo hr.retencion.renta
- Crear vistas list y form para cada modelo con filtros y orden

4. LÓGICA DE CÁLCULO DE RETENCIONES

Responsable: Backend

Tareas detalladas:

- Heredar el método compute_sheet() en hr.payslip
- Obtener salario mensual desde contract_id.wage
- Calcular:
 - ISSS: aplicar porcentaje a salario (hasta techo)
 - Comparar el salario mensual del empleado con el techo establecido para ISSS.
 - Si el salario excede el techo, usar el techo como base de cálculo.
 - Aplicar el porcentaje correspondiente al empleado sobre esa base.
 - Resultado: monto a retener por ISSS.

- AFP: aplicar porcentaje a salario (hasta techo)
 - Comparar el salario mensual con el techo establecido para AFP.
 - Si el salario excede el techo, usar el techo como base de cálculo.
 - Aplicar el porcentaje correspondiente al empleado sobre esa base.
 - Resultado: monto a retener por AFP.
- Renta: buscar tramo y calcular según fórmula
 - Determinar en qué tramo de la tabla de retención de renta cae el salario mensual.
 - Calcular el exceso del salario sobre el mínimo del tramo.
 - Aplicar el porcentaje del tramo sobre ese exceso.
 - Sumar la cuota fija del tramo.
 - Resultado: monto total a retener por renta.
- Agregar retenciones como input_line_ids (negativos)
- Validar que los valores se reflejen correctamente en el neto

5. ASIGNACIONES DINÁMICAS DE SALARIO

Responsable: Backend + Frontend

Tareas detalladas:

- Crear modelo hr.salary.assignment con campos:
 - employee_id (Many2one)
 - tipo (Selection: hora extra, comisión, etc.)
 - monto (Float)
 - periodo (Date o relación con payslip_id)
- Agregar relación inversa desde hr.employee
- Agregar relación con hr.payslip si se desea histórico
- Crear vista formulario con selección de tipo y monto

6. MENÚ DE GESTIÓN DE ASIGNACIONES

Responsable: Frontend

Tareas detalladas:

- Crear menú superior: Nómina > Asignaciones de Salario
- Crear vista list con:
 - columnas: empleado, tipo, monto, fecha
 - filtros por período, tipo o empleado
- Crear vista form para crear/editar asignaciones
- Opción de carga múltiple (opcional)

7. REPORTE DE NÓMINA

Responsable: Frontend

Tareas detalladas:

- Heredar plantilla QWeb: hr_payslip.report_payslip
- Agregar sección de:
 - Retenciones (ISSS, AFP, Renta) — mostrar nombre y valor
 - Asignaciones extra — mostrar tipo y monto
 - Total de ingresos, total deducciones, neto a pagar
- Usar t-foreach="slip.input_line_ids" para mostrar valores

8. PRUEBAS FUNCIONALES Y AJUSTES

Responsable: Backend + Frontend

Tareas detalladas:

- Crear contratos de prueba con diferentes salarios
- Validar cálculo automático de ISSS, AFP y Renta
- Probar rangos que exceden techos salariales
- Agregar varias asignaciones a un empleado y validar
- Confirmar que los montos se reflejen correctamente en el PDF

9. SEGURIDAD Y ACCESOS

Responsable: Backend

Tareas detalladas:

- En ir.model.access.csv dar acceso a modelos:
 - hr.retencion.renta
 - hr.retencion.isss
 - hr.retencion.afp
 - hr.salary.assignment
- Si aplica, crear grupos:
 - Grupo RRHH avanzado con permisos para editar configuraciones
- Probar con diferentes usuarios

Funcionalidad	Responsable	Tarea
1. Estructura inicial del módulo	Backend	Crear módulo Odoon base l10n_sv_hr_retenciones
	Backend	Configurar __manifest__.py con dependencias (hr, hr_payroll)
	Backend	Preparar estructura de carpetas: models/, views/, data/, security/
2. Modelos de configuración de retenciones	Backend	Crear modelo hr.retencion.iss: porcentaje, techo, aplicable a empleado/patrón
	Backend	Crear modelo hr.retencion.afp: porcentaje, techo, aplicable a empleado/patrón
	Backend	Crear modelo hr.retencion.renta: tabla escalonada con cuota fija, exceso y porcentaje
	Backend	Cargar datos de ejemplo en XML para cada tabla
3. Vista de configuración de retenciones	Frontend + Backend	Agregar nuevo menú: Configuración > Retenciones
	Frontend + Backend	Crear submenús: Tabla ISSS, Tabla AFP, Tabla Renta
	Frontend + Backend	Crear vistas list y form para cada tabla de retención
4. Lógica de cálculo de retenciones	Backend	Heredar hr.payslip.compute_sheet
	Backend	Obtener salario mensual desde contract_id.wage
	Backend	Calcular ISSS, AFP y Renta en base al salario base del contrato
	Backend	Agregar retenciones como input_line_ids (negativos)
	Backend	Validar que los valores se reflejen correctamente en el neto
5. Asignaciones dinámicas de salario	Backend + Frontend	Crear modelo hr.salary.assignment
	Backend + Frontend	Relacionar con hr.employee y hr.payslip
	Backend + Frontend	Crear vista (menú separado) para ingresar asignaciones
6. Menú de gestión de asignaciones	Frontend	Nuevo menú: Nómina > Asignaciones de Salario
	Frontend	Crear vista list y form con filtros por período y empleado
	Frontend	Permitir registrar múltiples asignaciones por empleado antes de cada ciclo de nómina
7. Reporte de nómina	Frontend	Heredar plantilla QWeb del recibo (hr_payslip.report_payslip)
	Frontend	Agregar detalle de: retenciones aplicadas, asignaciones aplicadas, totales ajustados
8. Pruebas funcionales y ajustes	Backend + Frontend	Prueba de cálculo con diferentes rangos salariales
	Backend + Frontend	Verificar techo de ISSS/AFP
	Backend + Frontend	Validar retención de Renta según tabla
	Backend + Frontend	Probar asignaciones múltiples y combinadas
	Backend + Frontend	Validar visualización correcta en la UI y reporte
9. Seguridad y accesos	Backend	Agregar reglas de acceso (ir.model.access.csv)
	Backend	Definir grupos si se desea limitar acceso a configuraciones