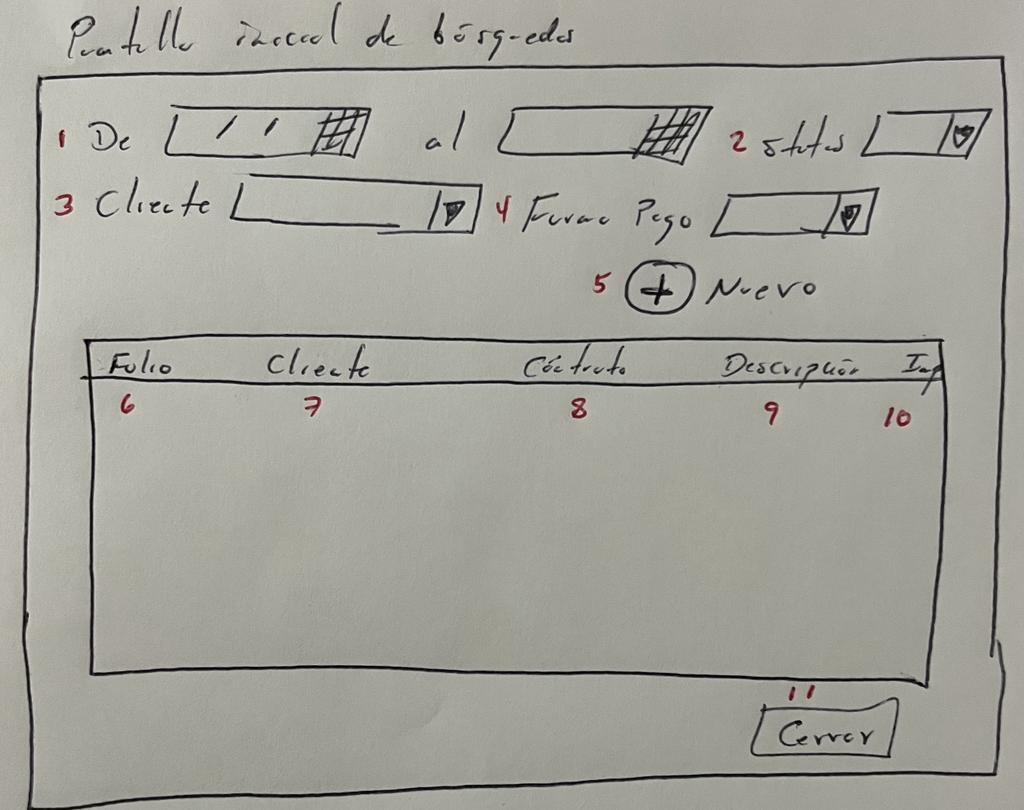
**PROCESO DE PAGOS**

Campos que van en la pantalla de búsqueda:

**Criterios de búsqueda:**

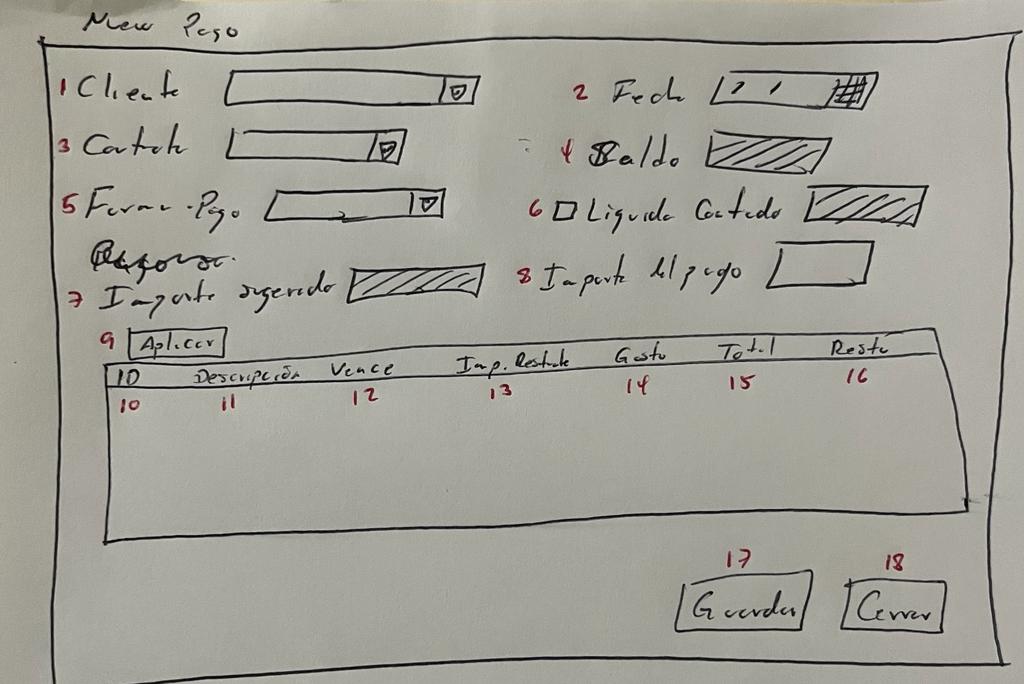


* **Por rango de fechas** (inicial y final): se muestran todos los pagos en ese rango de fechas, incluyendo las de las fechas inicial y final.
* **Status**: es a través de un combo, el usuario puede elegir el status que desea filtrar (vigentes / Cancelados / Todos). Por default, estará en Todos.
* **Cliente**: aquí busca y elige al cliente que se desea filtrar. Se muestran los pagos que se han hecho de ese cliente. Se pueden aplicar junto con otros filtros, como fecha y estatus.
* **Forma de pago**: en este combo se muestran las formas de pago, y se elige una de ellas. Se muestran los pagos que se han hecho con esa forma de pago. Se pueden aplicar junto con otros filtros, como fecha, estatus y cliente.
* **Nuevo**: se abre la pantalla de nuevo pago. Se explica más adelante.
* **Folio**: se muestra el ID de los pagos filtrados.
* **Cliente**. Se muestra el nombre de los clientes de los pagos filtrados.
* **Contrato**. Se muestra folio de los contratos de los pagos filtrados.
* **Descripción**. Se muestra la descripción de los pagos filtrados.
* **Importe**. Se muestra el importe total de los pagos filtrados.
* **\*\* Status**. No viene en la imagen, pero si hay que poner el status (vigente, cancelado).

**También faltan los botones. Qué hacen los botones:**

* **Editar:** muestra el pago, y aquí SI está activado el botón de Guardar. Si se le da clic en Guarda, se vuelve a guardar el mismo registro con los cambios hechos. Hay que poner un mensaje poniendo lo siguiente: “Para que los cambios tengan impacto en los saldos del contrato, se deberán reprocesar los pagos en la sección de Control de Contratos”.
* **Cancelar:** Se cambia el status a cancelada.

**Nuevo Pago:**

****

**\*\* Hay que poner un combo que sería el item5b:** con la etiqueta de Lugar, y el combo mostrar: Oficina, cliente, domicilio

1. **Cliente.** Más que nada es una búsqueda para buscar y elegir al cliente. NOTA: los clientes mostrados deben tener contratos vigentes.
2. **Fecha.** Por default, se pondrá la fecha actual, sin embargo, se le puede poner otra fecha. Se puede abrir un control para elegir la fecha deseada.
3. **Contrato**. Se mostrará un combo con los contratos vigentes del cliente elegido en el punto 1.
4. **Saldo.** Se muestra el valor del campo Contratos.restante.
5. **Forma de pago.** En este combo se muestra las descripciones de las formas de pago para que el usuario elija una.
6. **Liquida contado.** Esta opción cuanta con un cuadro check. En el cual, si se elige se activa el textBox de la derecha, en el cual si

Fecha\_item2 – Contratos.fecha <= 30 días, entonces:

En el textBox mostraría el siguiente importe:

textBox = Contratos.importePagoContado – Contratos.abonos

Pero, si MAYOR de 30 días, entonces mostrará el valor del campo Contratos.restante.

NOTA: este textBox no es editable.

1. **Importe sugerido.** En este textBox, si está activada el check de Liquida contado, entonces muestra ese mismo importe. Si no, entonces mostrará:
   1. Se toma el primer pagaré que NO está pagado de ese contrato.

Si (pagare.fecha < Fecha\_item2), entonces guardaremos en una variable a la que llamaremos impTemp = pagares.resta + sucursal.impCobranza

Se analiza el siguiente pagaré, y se vuelve aplicar la condición anterior, hasta que se deje de cumplir la condición de las fechas y se ACUMULA el impTemp. Al final se muestra en el textBox el importe de la variable impTemp.

* 1. Si es la PRIMERA VEZ que se checa la condición de las fechas, y si pagare.fecha >= Fecha\_item2), entonces impTemp = pagares.resta.

NOTA: este textBox no es editable.

1. **Importe del pago.** Aquí se teclea el importe del pago y éste debe ser mayor que CERO.

Si (importe\_item8 = importe\_item7), entonces se muestran en la tabla inferior, los pagarés afectados en el inciso anterior, desglosando los gastos de cobranza en cado de haber.

Si (importe\_item8 < importe\_item7), entonces se muestran en la tabla inferior, uno a uno los pagarés que a los que se le puede aplicar el importe restante, al campo restante de cada pagaré, junto con los gastos de cobranza, si es que aplica.

Si (importe\_item8 > importe\_item7). Explicación en el siguiente punto (9).

1. **Aplicar.** En la parte del guardado en la tabla de Detalle\_pago, se aprovecha este punto para explicar mejor.

**17. Guardar**

En la tabla de **Pagos**, se guarda:

|  |  |
| --- | --- |
| **PAGO** | |
| **Campo** | **¿Qué se guarda?** |
| ID | = se auto incrementa |
| Fecha | = fecha del ítem2 |
| Importe | = item8 |
| Lugar | = item5b (1: oficina, 2: cliente, 3: domicilio) |
| Observaciones | = null |
| Status | = True (vigente) |
| usuario | = ID del usuario logueado o del cobrador en caso de ser domicilio |
| contrato | = ID del contrato elegido |
| sucursal | = ID de la sucursal |
| formaPago | = ID de la forma de pago elegida |

En la tabla **Detalle\_pago** se guarda:

Aquí vamos a crear varias variables que usaremos de modo repetitivo mientras el importe del pago sea positivo:

impTemp = importe del ítem8 (importe del pago)

importeDet (se va a calcular)

nuevoImporte (se va a calcular)

impGasto (se va a calcular)

diferencia (se va a calcular)

impPagare (se va a calcular)

MIENTRAS (impTemp > 0)

{ Ubico al pagaré más viejo cuyo status = 1 (no pagado), del contrato que se seleccionó)

Si (pagare.fecha >= Fecha\_item2)

{ nuevoImporte = Pagare.resta;

Si (impTemp <= nuevoImporte)

importeDet = impTemp;

else

{ importeDet = nuevoImporte;

impTemp = impTemp – nuevoImporte;

}

impPagare = importeDet + importeDet;

impGasto = 0;

Contratos.abonos = Contratos.abonos + importeDet;

Contratos.resta = Contratos.resta – importeDet;

Pagare.importePagado = Pagare.importePagado + importeDet;

Pagare.resta= Pagare.resta – importeDet;

Pagare.fechaUltimoPago = Fecha\_item2;

Si (Pagare.resta < 1.0)

Pagare.status = 2 (pagado);

Y se guarda lo siguiente en Detalle\_pago

|  |  |
| --- | --- |
| **DETALLE\_PAGO** | |
| **Campo** | **¿Qué se guarda?** |
| ID | = auto incremento |
| pago | = Pago.ID |
| pagare | = Pagare.ID |
| importe | = importeDet |
| importePagare | = impPagare |
| importeGasto | = impGasto |
| Status | = True |

}

else // es el else del Si (pagare.fecha >= Fecha\_item2)

{ Si (Pagare.gastoCobranza = 0)

{nuevoImporte = Pagare.resta + Sucursal.impCobranza;

Pagare.gastoCobranza = Sucursal.impCobranza;

Pagare.total = Pagare.importeOriginal + Pagare.gastoCobranza;

Contrato.restante = Contrato.restante + Pagare.gastoCobranza;

}

else

nuevoImporte = Pagare.resta;

Si (impTemp <= nuevoImporte)

{ importeDet = impTemp;

Pagare.importePagado = Pagare.importePagado + importeDet;

Pagare.resta= Pagare.resta – importeDet;

Si ( Pagare. importePagado >= Pagare.importeOriginal )

{ impGasto = importeDet;

}

else

{ Si ( (impTemp + Pagare.importePagado) <= Pagare.importeOriginal)

{ impGasto = 0;

impPagare=importeDet; -

}

else

{ impPagare= Pagare.importeOriginal;

impGasto = impTemp + Pagare.importePagado – impPagare;

}

}

}

else // es el else del Si (impTemp <= nuevoImporte)

{ importeDet = nuevoImporte;

If(nuevoImporte < = Pagare.gastoCobranza)

{ impGasto = nuevoImporte;

}

Else

{ impPagare= Pagare.importeOriginal;

impGasto = impTemp + Pagare.importePagado – impPagare;

}

impTemp = impTemp – nuevoImporte;

}

Contratos.abonos = Contratos.abonos + importeDet;

Contratos.resta = Contratos.resta – importeDet;

Pagare.fechaUltimoPago = Fecha\_item2;

Y se guarda lo siguiente en Detalle\_pago

|  |  |
| --- | --- |
| **DETALLE\_PAGO** | |
| **Campo** | **¿Qué se guarda?** |
| ID | = auto incremento |
| pago | = Pago.ID |
| pagare | = Pagare.ID |
| importe | = importeDet |
| importePagare | = impPagare |
| importeGasto | = impGasto |
| Status | = True |

}