INFORME SERVICIOS DE HONORARIOS.

NOVIEMBRE	2025
Mes	**Año**

I.- Datos.

1.- Nombre.	CARLOS ALBERTO GONZALEZ ZAMORA		
2.- Rut.			
3.- Nº Decreto.	Alcaldicio: **835**	Imputación: **215.21.04.004.001**	Centro de Costos: **25.04.50**
4.- Dirección a la que pertenece.	**DIRECCION DE DESARROLLO COMUNITARIO -** *APOYO PARA LA MATERIALIZACIÓN DE ACCIONES SOCIALES Y COMUNITARIAS*		

II.- Función según lo que indica el contrato de prestación de servicios.

- **APOYO ADMINISTRATIVO PROGRAMA APOYO PARA LA MATERIALIZACIÓN DE ACCIONES SOCIALES Y COMUNITARIAS**
- Administración plataforma digital de gestión de las oficinas de DIDECO.
- Desarrollo y/o implementación de sistemas informáticos para la DIDECO.
- Gestión interna de las actividades estipuladas en los programas de cada oficina de DIDECO.

III.- Descripción de las funciones realizadas.

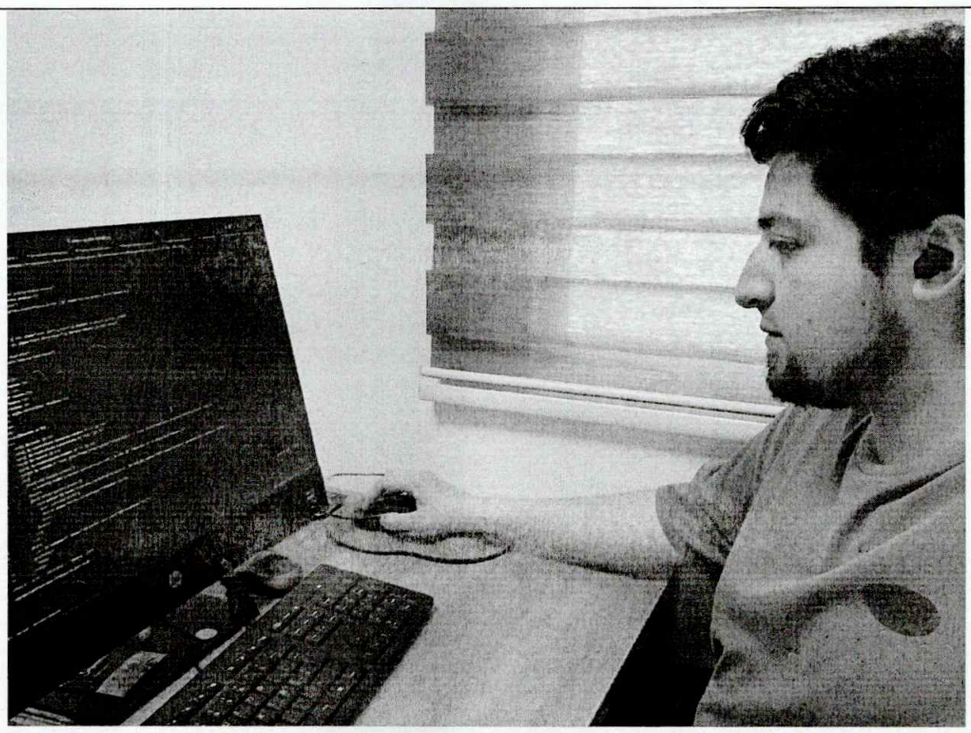
Se realizan avance en el sistema de eDideco, en el modulo de presupuesto
Se da asistencia a funcionarias en sus puestos de trabajo
Se realiza mantencion de impresora
Se da soporte a impresa que en varias ocaciones trababa hojas, aun se espera que lleguen a repararla adecuadamente
Se ayuda a funcionarios en sus puestos de trabajo
Se realizan avances al sistema de presupuestos de eDideco
Se continuan los avances con el sistema de edideco
Se genera un apartado al practicante para que pueda realizar sus labores de forma independiente
Se optimizan partes del codigo anterior para permitir mejoras mas fluidas
Se realiza un cambio de una luz que presentaba problemas
Se realizan avances en el sistema de eDideco
Se plantean realizar ajustes a la generacion de documentos e informes para tener mayor cantidad de opciones para los funcionarios
Se realizan avances en el sistema de eDideco
Se da apoyo a funcionarios en sus puestos de trabajo
Se generan solicitudes de contrato a funcionarios
Se reimprimen informes de honorarios que fueron solicitados
Se generan nuevas vistas para la intranet eDideco, para el apartado de las cuentas de los programas
Se realizan cambios menores en controladores para mejorar la legibilidad del codigo
Se da apoyo a funcionarios de Dideco
Se realiza una revision en un computador del programa de discapacidad
Se realizan avances con el sistema de intranet eDideco
Se realizan avances en eDideco
Se habilita plataforma para alumno en practica para que pueda estudiar el ambiente de desarollo en el cual se ha estado avanzando en la intranet eDideco
Se le da apoyo a funcionarios en sus puestos de trabajo

Se revisa computador en centro historico, presenta fallas en la tarjeta de video y debe ser reemplazado
Se realza cambio a tonner de impresora
Se realizan avances al sistema de eDideco
Se revisa la nueva ubicacion del programa OLN y se planifica la forma mas rapida para tener los puestos de trabajo operativos
Se recuperan las camaras que estaban instaladas corrigiendo problemas en el hardware que controlaba el sistema de camaras
Se revisan puntos de red habilitados al momento de la llegada del programa
Se etiquetan y marcan cables de red del edificio del programa OLN
Se realizan modificaciones a las nuevas vistas de eDideco

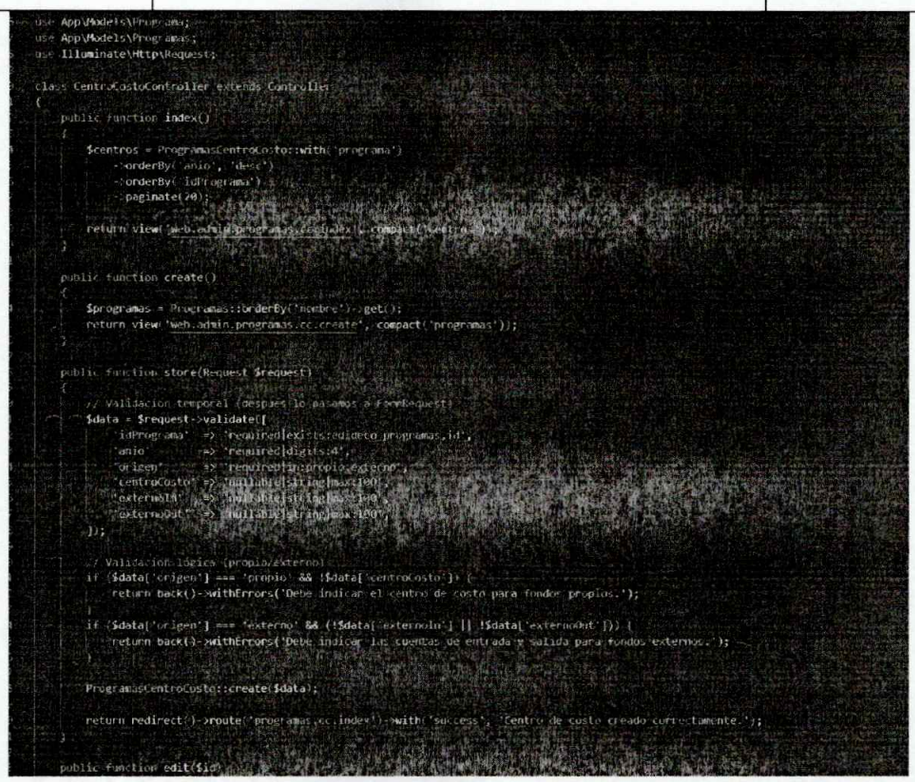
Se instalan puntos de internet inalambrico para los programas de PRLAC y OLN
Se revisa el cableado del edificio nuevo y se identifican para su manejo posterior
Se instala impresora de forma local en computadores de funcionarios que lo requerian
Se revisa falla en computador de Omil
Se realizan avances en eDideco, se modifican vistas para formularios y sus respectivos controladores
Se realiza mantencion en impresoras de funcionarios de Dideco
Se cambio cableado que presentaba problemas en oficina del Director, cable con fallas por daño
Se realiza la instalacion de nuevo chasis para totem en Dideco
Se realizan avances en eDideco
Se da apoyo a funcionarios en sus puestos de trabajo con problemas con archivos
Se realiza mantencion a computador que presentaba problemas en la oficina del programa de apoyo para la materializacion de acciones sociales y comunitarias
Se da mantencion a escanner que solia atascarse al pasar las hojas
Se realiza mantencion menor a impresora
Se realizan avances en sistema eDideco

IV.- Cobertura: (mencionar y adjuntar documentos de respaldo).

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-03	Fotografia en horario laboral	FOTOGRAFÍA



Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-04	Avances en el codigo	FOTOGRAFÍA



```
use App\Models\Programa;
use App\Models\Programas;
use Illuminate\Http\Request;

class CentroCostoController extends Controller
{
    public function index()
    {
        $centros = ProgramasCentroCosto::with('programa')
            ->orderBy('anio', 'desc')
            ->orderBy('idPrograma')
            ->paginate(20);

        return view('web.admin.programas.cc.index', compact('centros'));
    }

    public function create()
    {
        $programas = Programas::orderBy('nombre')->get();
        return view('web.admin.programas.cc.create', compact('programas'));
    }

    public function store(Request $request)
    {
        // Validación temporal (después lo pasamos a FormRequest)
        $data = $request->validate([
            'idPrograma'  => 'required|exists:mediacto_programas,id',
            'anio'        => 'required|digits:4',
            'origen'      => 'required|in:propio,externo',
            'centroCosto' => 'nullable|string|max:100',
            'externoIn'   => 'nullable|string|max:100',
            'externoOut'  => 'nullable|string|max:100',
        ]);

        // Validación lógica (propio/externo)
        if ($data['origen'] === 'propio' && !$data['centroCosto']) {
            return back()->withErrors('Debe indicar el centro de costo para fondos propios.');
        }

        if ($data['origen'] === 'externo' && (!$data['externoIn'] || !$data['externoOut'])) {
            return back()->withErrors('Debe indicar las cuentas de entrada y salida para fondos externos.');
        }

        ProgramasCentroCosto::create($data);

        return redirect()->route('programas.cc.index')->with('success', 'Centro de costo creado correctamente.');
    }

    public function edit($id)
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-05	modificaciones en vistas	FOTOGRAFÍA

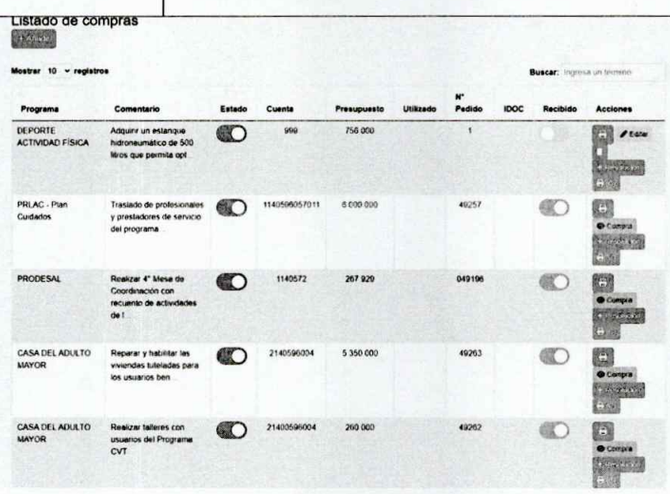
Evaluación de la Compra

Solo una evaluación por compra.

+ Crear Evaluación | Editar Evaluación

Nota: Crear una nueva evaluación reemplazará la existente.

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-05	modificaciones en vistas	FOTOGRAFÍA



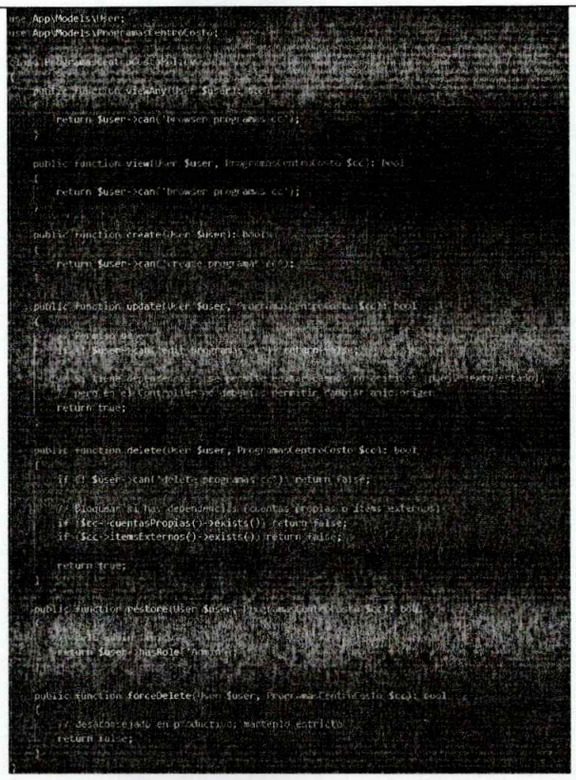
Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-05	Manejo de roles	FOTOGRAFÍA

Roles

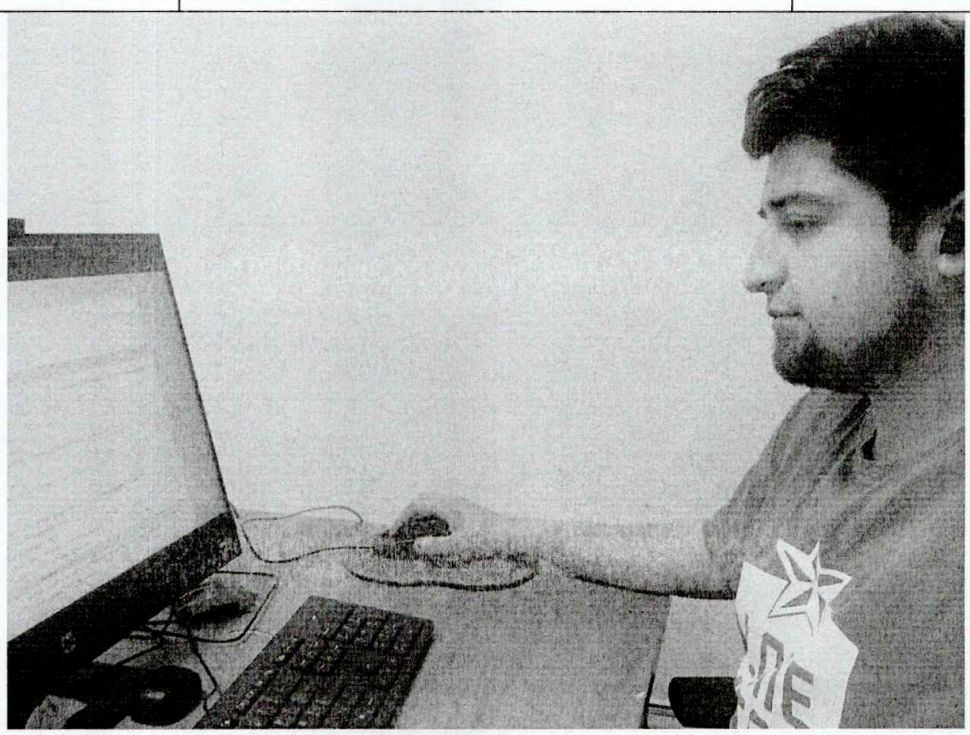
Crear Rol

ID	Nombre	Acciones
10	Admin	Editar Eliminar
11	user	Editar Eliminar
14	coordinador	Editar Eliminar
15	funcionario	Editar Eliminar
16	comprador	Editar Eliminar

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-06	Avances en el codigo	FOTOGRAFÍA

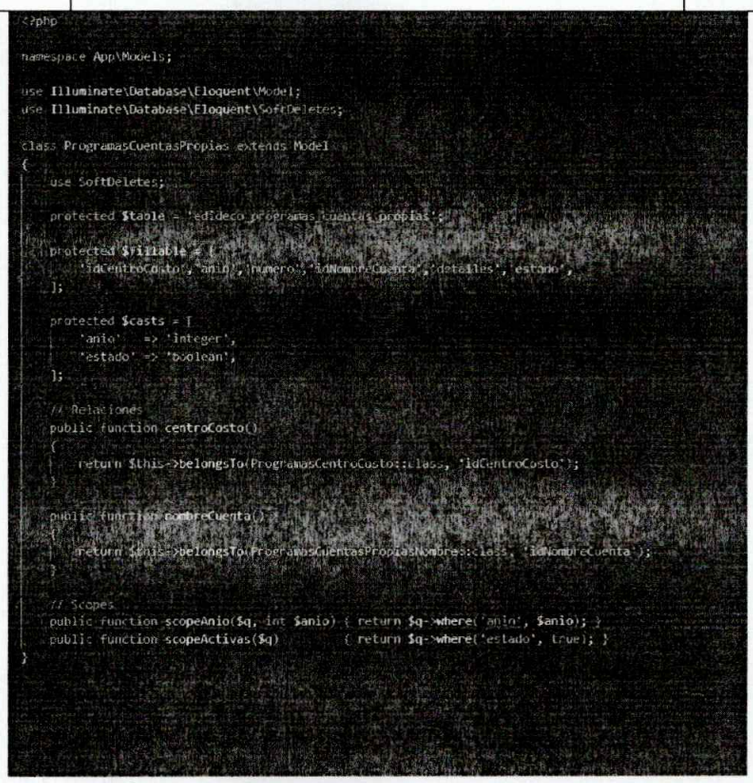


Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-07	Fotografia en horario laboral	FOTOGRAFÍA



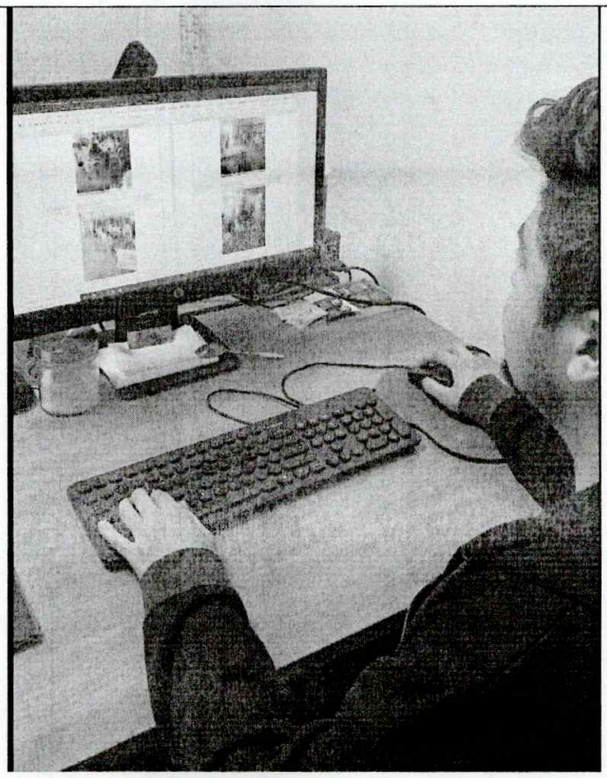
Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-10	modelos nuevos	FOTOGRAFÍA

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-10	modelos nuevos	FOTOGRAFÍA



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class ProgramasCuentasPropias extends Model
{
    use SoftDeletes;

    protected $table = 'edideco_programas_cuentas_propias';

    protected $fillable = [
        'IdCentroCosto', 'anio', 'numero', 'IdNombreCuenta', 'detalles', 'estado',
    ];

    protected $casts = [
        'anio'   => 'integer',
        'estado' => 'boolean',
    ];

    // Relaciones
    public function centroCosto()
    {
        return $this->belongsTo(ProgramasCentroCosto::class, 'IdCentroCosto');
    }

    public function nombreCuenta()
    {
        return $this->belongsTo(ProgramasCuentasPropiasNombre::class, 'IdNombreCuenta');
    }

    // Scopes
    public function scopeAnio($q, int $anio) { return $q->where('anio', $anio); }
    public function scopeActivas($q)         { return $q->where('estado', true); }
}
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-11	Fotografia en horario laboral	FOTOGRAFÍA



Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-12	Correo electronico	CORREO ELECTRÓNICO

Solicitud de correo y acceso eDIDECO Recibidos ×

Resumen

Diego Alfonso Pino Pavez lun, 10 nov, 11:27 (hace 11 días)
Estimado Director: Junto con saludar, me comunico con usted para gestionar, por favor, la creación de correo electrónico y acceso a eDIDECO para las siguientes

Jonathan Fernandez Figueroa lun, 10 nov, 11:33 (hace 11 días)
Estimados: Junto con saludar, solicito tengan a bien la creación de correos institucionales informados en correo de arrastre, que tienen relación a nuevos funci

Carlos González Zamora <czamora@buin.cl> mié, 12 nov, 10:49 (hace 9 días)
para Diego, Jonathan, Claudio, George

Estimado,

Junto con saludar, indico credenciales de acceso al sistema eDideco solicitadas

Ana Cartes Espinoza

- **Usuario:** Acartes@buin.cl
- **Contraseña:** 12345678

Jeannette Palomera Valdes

- **Usuario:** jeannette.palomera@buin.cl
- **Contraseña:** 12345678

Katherinne Orellana Sáez

- **Usuario:** korellana@buin.cl
- **Contraseña:** 12345678

El link de acceso es https://edideco.buin.cl
Para cambiar la contraseña, usar la opción "Mi perfil" al ingresar al sistema.

Saludos cordiales.

El lun, 10 nov 2025 a las 11:27, Diego Alfonso Pino Pavez (<dpavez@buin.cl>) escribió
...

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-12	Creacion de usuario	FOTOGRAFÍA

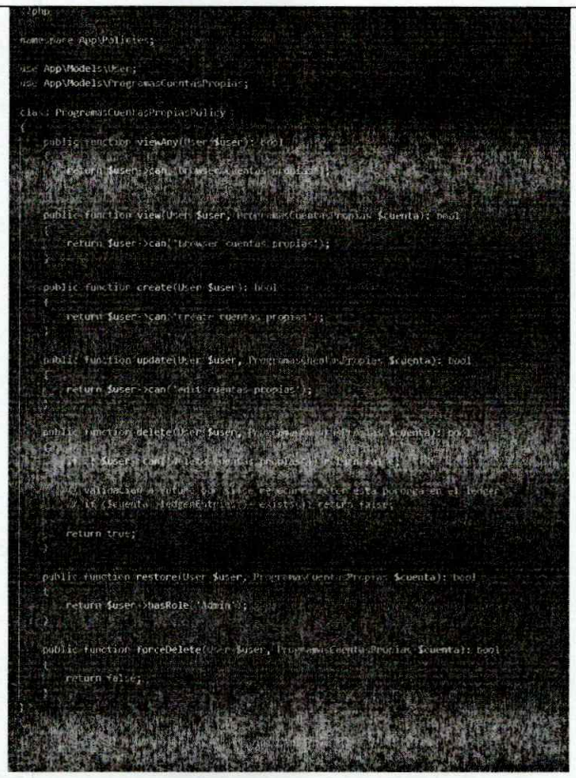
651	Ana Cartes Espinoza	Acartes@buin.cl	funcionario	Editar Eliminar
652	Jeannette Palomera Valdes	jeannette.palomera@buin.cl	funcionario	Editar Eliminar
653	Katherinne Orellana Saez	korellana@buin.cl	funcionario	Editar Eliminar
654	Claudia Fica	cfica@buin.cl	funcionario	Editar Eliminar
655	Maria Apablaza Orellana	angelesapablaza30@gmail.com	funcionario	Editar Eliminar

Mostrando 1 a 367 de 367 registros

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-13	Permisos nuevos	FOTOGRAFÍA

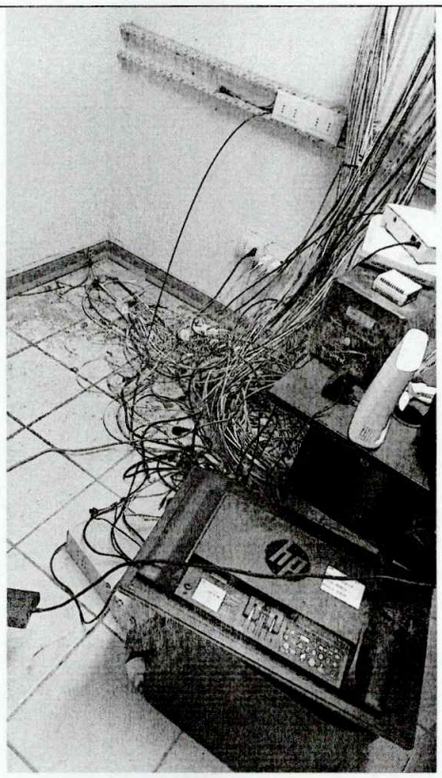
42	create programas cc	Editar	Eliminar
43	edit programas cc	Editar	Eliminar
44	delete programas cc	Editar	Eliminar
45	browser cuentas propias	Editar	Eliminar
46	create cuentas propias	Editar	Eliminar
47	edit cuentas propias	Editar	Eliminar
48	delete cuentas propias	Editar	Eliminar
49	browser cuentas propias nombres	Editar	Eliminar
50	create cuentas propias nombres	Editar	Eliminar
51	browser items externos	Editar	Eliminar
52	create items externos	Editar	Eliminar
53	edit items externos	Editar	Eliminar
54	delete items externos	Editar	Eliminar
55	browser cc items externos	Editar	Eliminar
56	create cc items externos	Editar	Eliminar

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-13	Avances en el codigo	FOTOGRAFÍA



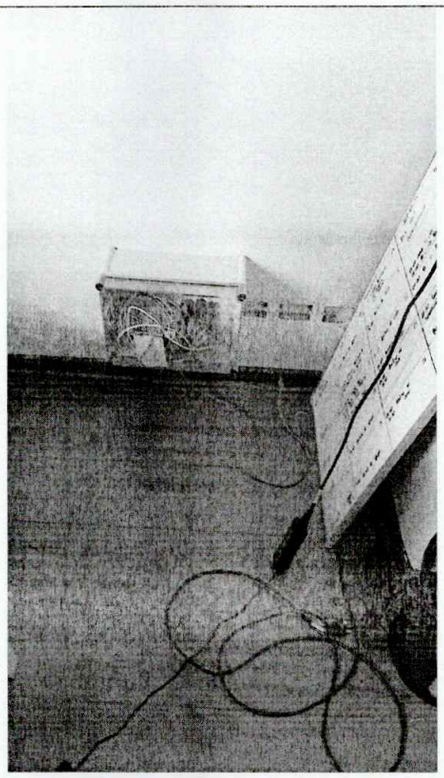
```
<?php

namespace App\Policies;

use App\Models\User;
use App\Models\ProgramasCuentasPropias;

class ProgramasCuentasPropiasPolicy
{
    public function viewAny(User $user): bool
    {
        return $user->can('browser cuentas propias');
    }

    public function view(User $user, ProgramasCuentasPropias $cuenta): bool
    {
        return $user->can('browser cuentas propias');
    }

    public function create(User $user): bool
    {
        return $user->can('create cuentas propias');
    }

    public function update(User $user, ProgramasCuentasPropias $cuenta): bool
    {
        return $user->can('edit cuentas propias');
    }

    public function delete(User $user, ProgramasCuentasPropias $cuenta): bool
    {
        return $user->can('delete cuentas propias');
        // validación adicional: no permitir eliminar si está protegido en el helper
        // if ($cuenta->esGenerado() || $cuenta->existe()) return false;

        return true;
    }

    public function restore(User $user, ProgramasCuentasPropias $cuenta): bool
    {
        return $user->hasRole('Admin');
    }

    public function forceDelete(User $user, ProgramasCuentasPropias $cuenta): bool
    {
        return false;
    }
}
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-14	Estado de cables de red del neuvo local	FOTOGRAFÍA

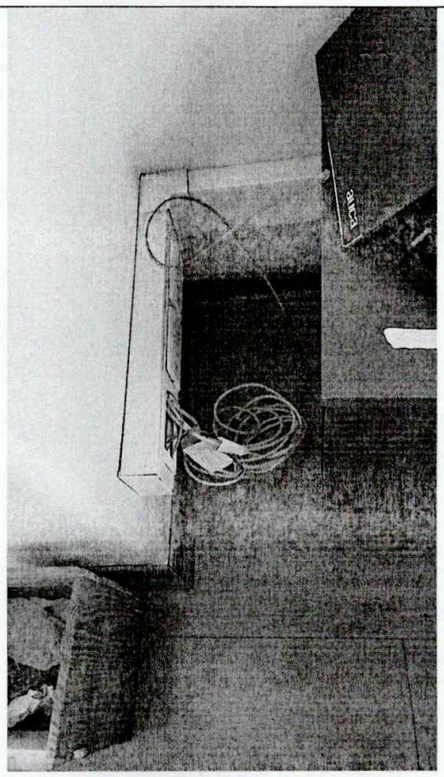


Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-17	revision, etiquetado y restauracion de cables	FOTOGRAFÍA

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-17	revision, etiquetado y restauracion de cables	FOTOGRAFÍA



Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-17	revision, etiquetado y restauracion de cables	FOTOGRAFÍA



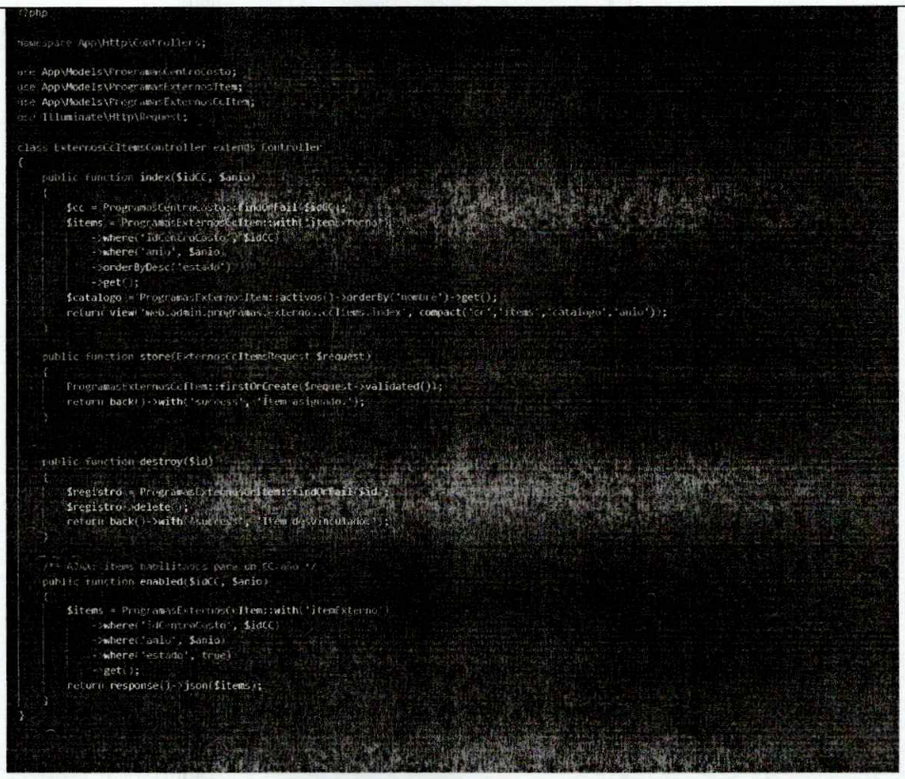
Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-18	Avances en el codigo	FOTOGRAFÍA

```php
namespace App\Http\Controllers;

use Illuminate\Foundation\Http\FormRequest;
use App\Models\ProgramasCentroCosto;

class CuentasPropiasRequest extends FormRequest
{
    public function authorize(): bool
    {
        return true;
    }

    public function rules(): array
    {
        $isCreate = $this->isMethod('post');

        return [
            'idCentroCosto'     => [$isCreate ? 'required' : 'nullable', 'exists:programas_centro_costos,id'],
            'anio'              => ['required', 'digits:4'],
            'numero'            => ['required', 'string', 'max:30'],
            'idNombreCuenta'    => ['required', 'exists:programas_cuentas_propias_nombres,id'],
            'detalles'          => ['nullable', 'string'],
            'estado'            => ['nullable', 'boolean'],
        ];
    }

    public function withValidator($validator)
    {
        $validator->after(function ($v) {
            $ccId = $this->input('idCentroCosto');
            if ($ccId) {
                $cc = ProgramasCentroCosto::find($ccId);

                if (!$cc) {
                    $v->errors()->add('idCentroCosto', 'El centro de costo seleccionado no existe.');
                    return;
                }

                if (!$cc->esPropio()) {
                    $v->errors()->add('idCentroCosto', 'El centro de costo debe ser de origen propio.');
                }

                // forzar que anio de cuenta = anio del CC
                if ((int)$this->anio !== (int)$cc->anio) {
                    $v->errors()->add('anio', 'El año de la cuenta no coincide con el del centro de costo.');
                }
            }
        });
    }
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-18	Avances en el codigo	FOTOGRAFÍA

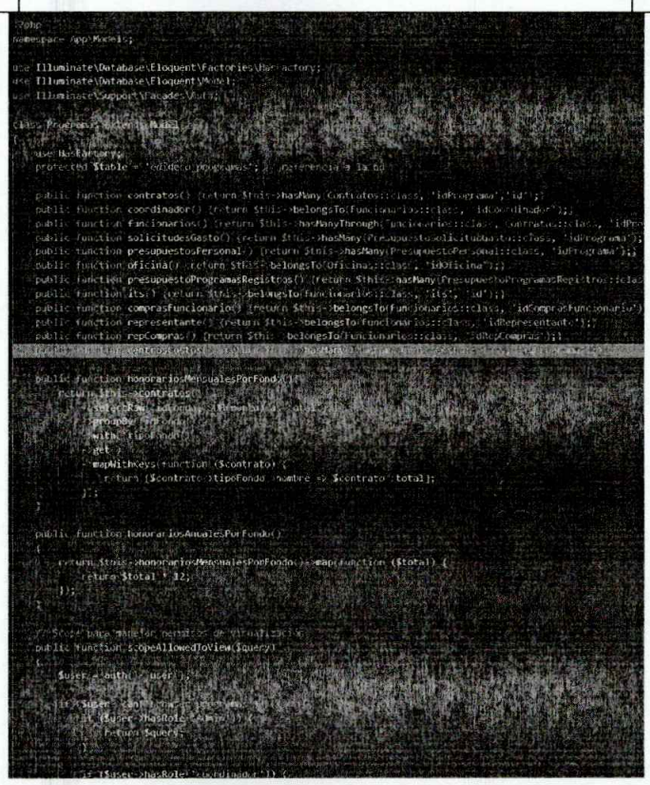
```php
namespace App\Http\Controllers;

use App\Models\Programas;
use App\Models\ProgramasCentroCosto;
use App\Models\ProgramasCuentasPropias;
use App\Models\ProgramasCuentasPropiasNombre;
use Illuminate\Http\Request;

class CuentasPropiasController extends Controller
{
    public function index()
    {
        $cuentas = ProgramasCuentasPropias::with['CentroCosto', 'nombre', 'programa']
            ->orderBy('anio', 'desc')
            ->paginate(20);
        return view('web.admin.programas.cuentaspropias.index', compact('cuentas'));
    }

    public function create()
    {
        $programas = Programas::where('estado', 1)->orderBy('nombre')->orderBy('version', 'desc')->get();
        $nombres = ProgramasCuentasPropiasNombre::orderBy('nombre')->get();

        return view('web.admin.programas.cuentaspropias.create', compact('programas', 'nombres'));
    }

    public function store(CuentasPropiasRequest $request)
    {
        ProgramasCuentasPropias::create($request->validated());
        return redirect()->route('programas.cuentaspropias.index')->with('success', 'Cuenta creada correctamente.');
    }

    public function edit($id)
    {
        $cuenta = ProgramasCuentasPropias::findOrFail($id);
        $nombres = ProgramasCuentasPropiasNombre::orderBy('nombre')->get();
        return view('web.admin.programas.cuentaspropias.edit', compact('cuenta', 'nombres'));
    }

    public function update(Request $request, $id)
    {
        $cuenta = ProgramasCuentasPropias::findOrFail($id);
        $data = $request->validate[
            'numero'          => 'required|string|max:30',
            'idNombreCuenta'  => 'required|exists:mysql:dbo.programas_cuentas_propias_nombres,id',
            'detalles'        => 'nullable|string',
            'estado'          => 'boolean',
        ];
        $cuenta->update($data);
        return redirect()->route('programas.cuentaspropias.index')->with('success', 'Cuenta actualizada correctamente.');
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-19	Avances en el codigo	FOTOGRAFÍA



```php
<?php

namespace App\Http\Controllers;

use App\Models\ProgramasCentroCosto;
use App\Models\ProgramasExternosItem;
use App\Models\ProgramasExternosCcItem;
use Illuminate\Http\Request;

class ExternosCcItemsController extends Controller
{
    public function index($idCC, $anio)
    {
        $cc = ProgramasCentroCosto::findOrFail($idCC);
        $items = ProgramasExternosCcItem::with('itemExterno')
            ->where('IdCentroCosto', $idCC)
            ->where('anio', $anio)
            ->orderByDesc('estado')
            ->get();
        $catalogo = ProgramasExternosItem::activos()->orderBy('nombre')->get();
        return view('web.admin.programas.externos.ccItems.index', compact('cc','items','catalogo','anio'));
    }


    public function store(ExternosCcItemsRequest $request)
    {
        ProgramasExternosCcItem::firstOrCreate($request->validated());
        return back()->with('success', 'Item asignado.');
    }


    public function destroy($id)
    {
        $registro = ProgramasExternosCcItem::findOrFail($id);
        $registro->delete();
        return back()->with('success', 'Item desvinculado.');
    }

    /** Ajax: items habilitados para un CC+año */
    public function enabled($idCC, $anio)
    {
        $items = ProgramasExternosCcItem::with('itemExterno')
            ->where('IdCentroCosto', $idCC)
            ->where('anio', $anio)
            ->where('estado', true)
            ->get();
        return response()->json($items);
    }
}
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-19	Avances en el codigo	FOTOGRAFÍA



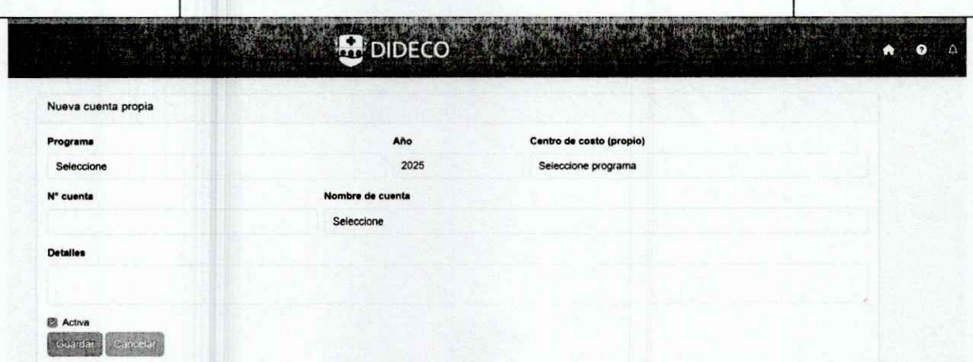
```php
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\Auth;

class ProgramasCentroCosto
{
    use HasFactory;
    protected $table = 'cidoco_programas'; // referencia a la BD

    public function contratos() {return $this->hasMany(Contratos::class, 'idPrograma','id');}
    public function coordinador() {return $this->belongsTo(Funcionarios::class, 'idCoordinador');}
    public function funcionarios() {return $this->hasManyThrough(Funcionarios::class, Contratos::class, 'idPro
    public function solicitudesGasto() {return $this->hasMany(PresupuestoSolicitudGasto::class, 'idPrograma');}
    public function presupuestosPersonal() {return $this->hasMany(PresupuestoPersonal::class, 'idPrograma');}
    public function oficina() {return $this->belongsTo(Oficina::class, 'idOficina');}
    public function presupuestoProgramasRegistros() {return $this->hasMany(PresupuestoProgramasRegistros::cla
    public function its() {return $this->belongsTo(Funcionarios::class, 'its', 'id');}
    public function comprasfuncionario() {return $this->belongsTo(Funcionarios::class, 'idCompraFuncionario')
    public function representante() {return $this->belongsTo(Funcionarios::class, 'idRepresentante');}
    public function repCompras() {return $this->belongsTo(Funcionarios::class, 'idRepCompras');}
    [illegible]

    public function honorariosMensualesPorFondo()
    {
        return $this->contratos()
            [illegible]
            ->groupBy
            [illegible]
            ->with('tipoFondo')
            ->get()
            ->mapWithKeys(function ($contrato) {
                return [$contrato->tipoFondo->nombre => $contrato->total];
            });
    }

    public function honorariosAnualesPorFondo()
    {
        return $this->honorariosMensualesPorFondo()->map(function ($total) {
            return $total * 12;
        });
    }

    // Scope base para permisos de visualización
    public function scopeAllowedToView($query)
    {
        $user = Auth::user();

        if ($user->hasRole('admin')) {
            return $query;
        }

        if ($user->hasRole('coordinador')) {
            return $query;
        }

        if ($user->hasRole('coordinador')) {
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-19	Avances en el codigo	FOTOGRAFÍA

```php
namespace App\Http\Controllers;

use Illuminate\Foundation\Http\FormRequest;
use App\Models\ProgramaCentroCosto;

class ExternosCtiumsRequest extends FormRequest
{
    public function authorize(): bool
    {
        return true;
    }

    public function rules(): array
    {
        $isCreate = $this->isMethod('post');

        return [
            'idCentroCosto' => [$isCreate ? 'required' : 'nullable', 'exists:dideco_programas_centro_costos,id'],
            'idItemExterno' => [$isCreate ? 'required' : 'nullable', 'exists:dideco_programas_externos_items,id'],
            'anio'          => ['required', 'digits:4'],
            'estado'        => ['nullable', 'boolean'],
        ];
    }

    public function withValidator($validator)
    {
        $validator->after(function ($v) {
            $cc = $this->input('idCentroCosto');

            if ($cc) {
                $ccModel = ProgramaCentroCosto::find($cc);
                if (!$ccModel || !$ccModel->esExterno()) {
                    $v->errors()->add('idCentroCosto', 'El centro de costo debe ser de origen externo.');
                }
            }
        });
    }

    public function messages(): array
    {
        return [
            'idCentroCosto.required' => 'Debe indicar el centro de costo.',
            'idItemExterno.required' => 'Debe seleccionar el ítem externo.',
            'anio.required'          => 'Debe indicar el año.',
        ];
    }
}
```

Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-20	Vistas nuevas	FOTOGRAFÍA

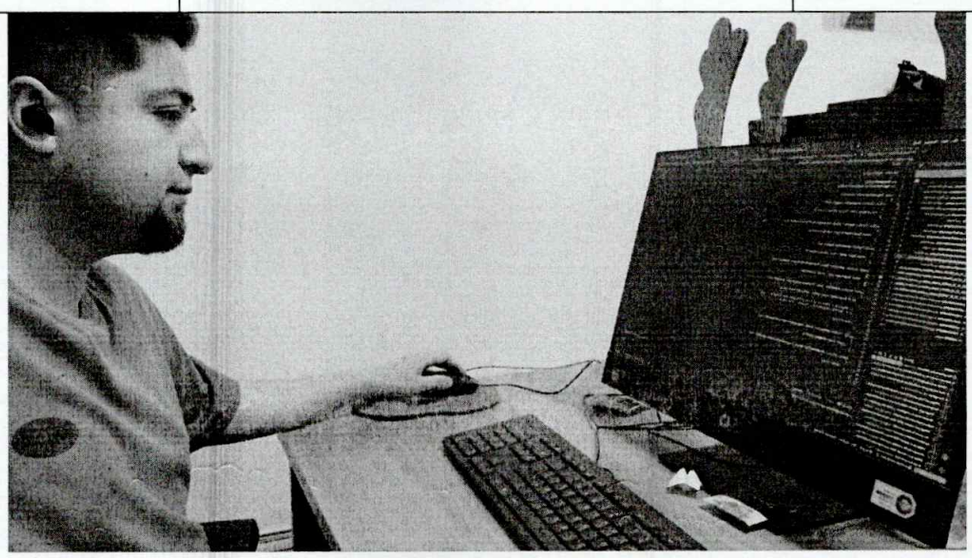
Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:
2025-11-20	Vistas nuevas	FOTOGRAFÍA



Fecha Cobertura:	Descripción Cobertura:	Tipo Cobertura:

2025-11-20	Cambio de totem	FOTOGRAFÍA

Fecha Cobertura: 2025-11-21	Descripción Cobertura: Fotografia en horario laboral	Tipo Cobertura: FOTOGRAFÍA



V.- Observaciones.

CERTIFICADO CONTRAPARTE TÉCNICA

Yo, **JONATHAN FERNÁNDEZ FIGUEROA**, **DIRECTOR DE DESARROLLO COMUNITARIO**, en mi calidad de contraparte técnica del prestador/a de servicios a honorarios que emite el presente informe, certifico que ha realizado sus funciones en el presente mes, en virtud al contrato de prestación de servicios que corresponde, y a los objetivos y requerimientos de las labores aprobadas por el concejo municipal o del Programa Municipal en que se desempeña.

JONATHAN FERNÁNDEZ FIGUEROA
DIRECTOR
DIRECCIÓN DE DESARROLLO COMUNITARIO

GEORGE OCTAVIO ELSO OSORIO
COORDINADOR (A) PROGRAMA