

Formularios Dinámicos - Documentación

Descripción General

El sistema de formularios dinámicos permite crear, gestionar y procesar formularios configurables que se adaptan automáticamente a los esquemas JSON esperados por el frontend. Esta implementación utiliza **arquitectura hexagonal** para mantener la separación de responsabilidades.

Arquitectura

```
src/forms/
├── domain/
│   ├── entities/           # Entidades del dominio
│   └── ports/              # Interfaces (puertos)
├── application/
│   └── services/           # Lógica de aplicación
└── infrastructure/
    ├── controllers/        # Controladores REST
    ├── entities/           # Entidades de TypeORM
    ├── repositories/       # Implementaciones de repositorios
    └── services/           # Servicios de infraestructura
```

Formularios Disponibles

1. Alta de Automotor (vehicle-registration)

- **Título:** Alta de Automotor
- **Formulario:** AUFA0030_CBA
- **Categoría:** vehiculos
- **Endpoint:** POST /automotor/transfereencia

Campos disponibles:

- Patente (text) - Validación de formato

- Tipo de Vehículo (select)
- Año de Fabricación (number)
- Origen RNPA (select)
- Color del Vehículo (select)
- CUIT Propietario (text) - Con validación
- Porcentaje de Propiedad (number)
- Es Responsable Inscripto (checkbox)
- Número de Motor (text)
- Número de Chasis (text)

2. Transferencia de Automotor (vehicle-transfer)

- **Título:** Transferencia de Automotor
- **Formulario:** AUFA0090_CBA - TAXIDESI
- **Categoría:** vehiculos
- **Endpoint:** PUT /automotor/transferencia

Campos disponibles:

- Patente del Vehículo (text)
- Fecha de Transferencia (date)
- Monto de la Operación (number)
- CUIT Vendedor (text)
- CUIT Comprador (text)
- Porcentaje de Propiedad (number)
- Es Responsable Inscripto (checkbox)

API Endpoints

Obtener Formularios

GET /forms

Retorna todos los formularios activos.

Respuesta:

```
[
  {
    "id": "vehicle-registration",
    "title": "Alta de Automotor",
    "description": "AUFA0030_CBA - Registre un nuevo vehículo en el sistema tributario",
    "category": "vehiculos",
    "isActive": true,
    "createdAt": "2025-07-17T14:03:03.040Z",
    "updatedAt": "2025-07-17T14:03:03.040Z"
  }
]
```

GET /forms/:id

Retorna un formulario específico con todos sus campos.

Ejemplo: GET /forms/vehicle-registration

Respuesta:

```

{
  "id": "vehicle-registration",
  "title": "Alta de Automotor",
  "subtitle": "Sistema de registro de nuevos vehículos - Formulario AUFA0030_CBA",
  "description": "AUFA0030_CBA - Registre un nuevo vehículo en el sistema tributario",
  "category": "vehiculos",
  "isActive": true,
  "version": "1.0",
  "buttonsConfig": {
    "submit": "Registrar Vehículo",
    "cancel": "Cancelar",
    "showBack": true,
    "backRoute": "/home"
  },
  "layoutConfig": {
    "cols": 2,
    "gap": "16px",
    "rowHeight": "85px"
  },
  "submissionEndpoint": "http://localhost:3000/automotor/transferencia",
  "submissionMethod": "POST",
  "sectionsConfig": [
    {
      "title": "Datos Básicos del Vehículo",
      "fields": ["patente", "tipo_vehiculo", "anio_fabricacion", "origen_rnpa", "color_vehiculo"]
    }
  ],
  "fields": [
    {
      "id": "uuid",
      "name": "patente",
      "type": "text",
      "label": "Patente",
      "placeholder": "AA123BB o ABC123",
      "required": true,
      "validationConfig": "...",
      "eventsConfig": "...",
      "orderIndex": 1
    }
  ]
}

```

GET /forms/category/:category

Retorna formularios por categoría.

Ejemplo: GET /forms/category/vehiculos

Envío de Formularios

POST /forms/:id/submit

Envía un formulario con transformación automática de datos.

Ejemplo: POST /forms/vehicle-registration/submit

Request Body:

```
{
  "patente": "AA123BB",
  "tipo_vehiculo": "AUTO",
  "anio_fabricacion": 2020,
  "cuit_propietario": "20-12345678-9",
  "porcentaje_propiedad": 100,
  "es_responsable": true,
  "numero_motor": "ABC123456",
  "numero_chasis": "DEF789012"
}
```

Transformación automática según `submissionSchema` :

```
{
  "ovpId": "AA123BB",
  "tipoOperacion": "ALTA",
  "propietario": {
    "cuit": "20-12345678-9",
    "porcentajePropiedad": 100,
    "esResponsable": true
  },
  "vehiculo": {
    "patente": "AA123BB",
    "anioFabricacion": 2020,
    "numeroMotor": "ABC123456",
    "numeroChasis": "DEF789012"
  }
}
```

Validación

POST /forms/validate

Valida datos de formulario sin enviar.

Request Body:

```
{
  "formId": "vehicle-registration",
  "data": {
    "patente": "AA123BB",
    "cuit_propietario": "20-12345678-9"
  }
}
```

Utilidades

POST /forms/autocomplete

Obtiene datos de autocompletado para campos específicos.

Request Body:

```
{
  "field": "patente",
  "query": "AA123",
  "formId": "vehicle-registration"
}
```

POST /forms/dependent-options

Obtiene opciones dependientes basadas en otros campos.

Request Body:

```
{
  "field": "modelo",
  "dependencies": {
    "marca": "FORD"
  },
  "formId": "vehicle-registration"
}
```

Configuración de Campos

Tipos de Campo Soportados

- text : Campos de texto
- number : Campos numéricos
- date : Selectores de fecha
- select : Listas desplegables
- checkbox : Casillas de verificación
- radio : Botones de radio

Validaciones

Cada campo puede tener reglas de validación en `validationConfig` :

```
{
  "pattern": "^[A-Z]{2}\\d{3}[A-Z]{2}$",
  "message": "Formato de patente inválido",
  "min": 1900,
  "max": 2025
}
```

Eventos

Los campos pueden tener eventos configurados en `eventsConfig` :

```
{
  "onBlur": "patenteLookup",
  "onChange": "calculateValue"
}
```



Base de Datos

Tabla `forms`

Almacena la configuración de formularios.

Tabla `form_fields`

Almacena los campos de cada formulario con su configuración.

Tabla `form_submissions`

Almacena los envíos de formularios para auditoría.



Inicialización

Los formularios se pueblan automáticamente al iniciar la aplicación mediante:

- `FormSeedService` : Crea los formularios base
- `FormFieldSeedService` : Crea los campos de los formularios

Integración con Frontend

El sistema está diseñado para ser consumido por frontends que esperan:

1. Configuración dinámica de formularios
2. Validación en tiempo real
3. Transformación automática de datos
4. Eventos de autocompletado y validación

Desarrollo

Para agregar nuevos formularios:

1. **Crear formulario** en `FormSeedService`
2. **Definir campos** en `FormFieldSeedService`
3. **Configurar transformación** en `submissionSchema`
4. **Agregar validaciones** específicas
5. **Testear endpoints** de integración

Notas Técnicas

- Los formularios usan **UUID** como identificadores
- Las configuraciones se almacenan como **JSON strings**
- La validación se ejecuta tanto en frontend como backend
- Los datos se transforman automáticamente antes del envío al endpoint destino
- El sistema es extensible para agregar nuevos tipos de campo y validaciones