

Informe de Avances - Entrega 1: CRUD de Tres Entidades Principales

1. Objetivo y Cumplimiento

El objetivo de esta entrega fue construir la base del sistema implementando las operaciones CRUD completas para las tres entidades principales: **Programa Académico**, **Periodo Académico** y **Asignatura**.

El sistema ha alcanzado el 100% de la funcionalidad requerida, incluyendo la creación de migraciones, operaciones CRUD completas y validaciones básicas.

Entidad	Operaciones CRUD	Validaciones Básicas	Estado de la Implementación
Programa Académico	Completo	Implementadas (obligatoriedad, unicidad, longitud)	Terminado
Periodo Académico	Completo	Implementadas (obligatoriedad, fechas válidas, lógica de estado)	Terminado
Asignatura	Completo	Implementadas (unicidad de nombre, carga horaria, tipo)	Terminado

Cumplimiento de Criterios de Aceptación (AC):

- **AC1 & Checklist:** Las **migraciones** para las tres entidades (**programas**, **periodos**, **asignaturas**) fueron implementadas y ejecutadas.
- **AC2 & Checklist:** El **CRUD completo** funciona sin errores para las tres entidades.
- **AC3 & Checklist:** Las **validaciones básicas** están implementadas en la capa de Aplicación/Dominio.
- **AC4 & Checklist:** El **video demostrativo** y la **documentación en Bruno** han sido completados.

2. Decisiones de Diseño y Arquitectura

El sistema está organizado bajo el patrón **Arquitectura Hexagonal (Puertos y Adaptadores)**, utilizando **TypeScript** con Fastify y **PostgreSQL**. Esta estructura permite la clara separación de responsabilidades:

Capa	Rol Principal
Dominio	Define las Entidades y las Reglas de Negocio . Contiene las interfaces de Repositorio (Puertos).
Aplicación	Contiene los Casos de Uso (Crear, Listar, Actualizar, Eliminar)[cite: 11]. Transforma datos entre la Entidad y el Tipo Plano.
Infraestructura	Implementa los Repositorios de PostgreSQL (Adaptadores) y el manejo de la conexión[cite: 9].
Presentación	Define las Rutas HTTP (Fastify) , recibe peticiones y devuelve respuestas JSON[cite: 11].

Persistencia (Inversión de Dependencias): Se mantiene el principio de que los **Repositorios de Infraestructura** solo manejan **Tipos Planos** (`IPeriodoAcademico`, `IProgramaAcademico`, etc.). La **Capa de Aplicación** es la responsable de instanciar las Clases Entidad para aplicar la lógica de negocio antes de la persistencia, asegurando el desacoplamiento de la base de datos de las reglas del negocio.

3. Reglas de Negocio Implementadas por Entidad

3.1. Programa Académico

- **Funcionalidad:** API REST para crear, ver, actualizar y eliminar programas académicos.
- **Propiedades:** ID (UUID), Nombre, Descripción, Nivel, Modalidad, Duración (valor y unidad).
- **Reglas y Validaciones:**
 - `Nombre` y `Descripción` son obligatorios.
 - La duración debe ser un valor positivo.

- La actualización mediante PUT está limitada a nombre y descripción (los campos estructurales como nivel no se modifican).

3.2. Periodo Académico

- **Funcionalidad:** Gestión del ciclo académico (nombre, fechas de inicio y fin, estado actual)⁹.
- **Lógica de Dominio:** Implementada la función de dominio actualizarEstado() en la Entidad, que automáticamente clasifica el periodo como 'activo', 'inactivo' o 'cerrado' basándose en las fechas definidas.
- **Validaciones:**
 - El nombre del periodo debe ser único (unicidad simple).
 - fechaFin debe ser posterior a fechaInicio.

3.3. Asignatura

- **Funcionalidad:** Gestión de materias impartidas (carga horaria, tipo)¹⁰.
- **Validaciones:**
 - **Unicidad Simple:** Implementada la validación para asegurar que no se pueda registrar una asignatura con un nombre ya existente en el sistema.
 - Carga horaria y tipo (teórica, práctica o mixta) son obligatorios.

4. Documentación y Entregables

- **Documentación Bruno:** Todos los endpoints CRUD (POST, GET, PUT, DELETE) para las tres entidades (/programas-academicos, /periodos, /asignaturas) han sido documentados en la pestaña Docs de Bruno, utilizando Markdown para describir el cuerpo de la petición, la respuesta y las validaciones aplicadas.
- **Entregables Finales:** El informe, el video demostrativo y el Pull Request desde release/sprint-1 hacia main se han completado según el checklist final.