INSTITUTIO SUPERIOR SUDAMERICANO

NOMBRE:

Paula Verdugo

ASIGNATURA:

Despliegue de

diagramas

DOCENTE:

Ing.Vacacela Jhostin

CURSO:

Tercer ciclo de software

CURSO:

2025-2026

Introducción

En los sistemas informáticos utilizados por las instituciones universitarias, la correcta administración de la información es fundamental, ya que se gestionan datos críticos relacionados con estudiantes, asignaturas y procesos académicos.

Para asegurar la integridad, consistencia y confiabilidad de esta información, es indispensable el uso de mecanismos que protejan las operaciones realizadas sobre la base de datos.

En este contexto, los principios ACID cumplen un rol esencial, pues garantizan que cada transacción se ejecute de manera segura y controlada, evitando errores o pérdidas de información.

Dentro de este proyecto, dichos principios se aplican especialmente en el proceso de matriculación estudiantil, donde es necesario asegurar que todas las acciones se completen correctamente o se reviertan en caso de fallos.

La Atomicidad:

La atomicidad establece que una transacción debe ejecutarse como una sola unidad indivisible, lo que implica que todas sus acciones se completan exitosamente o ninguna se aplica.
Durante el proceso de inscripción académica, el sistema lleva a cabo varias validaciones y registros de manera secuencial, entre ellas:

- Comprueba que el estudiante se encuentre habilitado para matricularse
- Confirma la disponibilidad de espacios en la asignatura
- Guarda el registro de la inscripción
- Actualiza la cantidad de cupos restantes

Si ocurre algún inconveniente en cualquiera de estas etapas, el sistema revierte automáticamente toda la operación. De esta forma, se garantiza que no existan registros inconsistentes ni errores relacionados con la disponibilidad de cupos.

La Consistencia:

La consistencia garantiza que la información almacenada en la base de datos respete todas las normas y restricciones definidas por el sistema. Esto permite que cada operación mantenga la validez de los datos antes y después de ejecutarse.

Dentro del proyecto, este principio se refleja en reglas claras como las siguientes:

- Solo los estudiantes habilitados pueden realizar procesos de matriculación
- Las asignaturas no pueden superar ni reducir incorrectamente su número de cupos disponibles
- El sistema impide registrar información inexistente o inválida

Gracias a la aplicación de estas restricciones, la base de datos siempre evoluciona de un estado íntegro a otro igualmente confiable, evitando inconsistencias y errores lógicos.

Aislamiento

El aislamiento se encarga de que las transacciones se ejecuten de manera independiente, evitando que una operación afecte a otra mientras está en proceso. Este principio resulta fundamental en entornos donde múltiples usuarios interactúan simultáneamente con el sistema.

En situaciones donde varios estudiantes intentan inscribirse al mismo tiempo en una misma asignatura, el sistema aplica mecanismos de control que permiten:

- Gestionar la disponibilidad de cupos de forma controlada y confiable
- Impedir que un mismo espacio sea asignado a más de un estudiante
- Preservar la coherencia y el orden de la información registrada

Gracias al aislamiento, el sistema puede manejar altos niveles de concurrencia sin comprometer la integridad de los datos.

Durabilidad

La durabilidad asegura que los cambios confirmados en el sistema permanezcan almacenados de manera definitiva, incluso ante fallos inesperados o interrupciones del servicio. Una vez que el proceso de matrícula se completa con éxito, la información registrada se mantiene protegida y disponible.

En este proyecto, este principio se refleja en los siguientes aspectos:

- Los datos de la matrícula quedan almacenados de forma permanente en la base de datos
- La información no se pierde, aun cuando ocurra un apagado del servidor o una falla del sistema
- El historial académico del estudiante se conserva de manera íntegra

Gracias a la durabilidad, se garantiza la confiabilidad y seguridad de la información académica a largo plazo.

Conclusión:

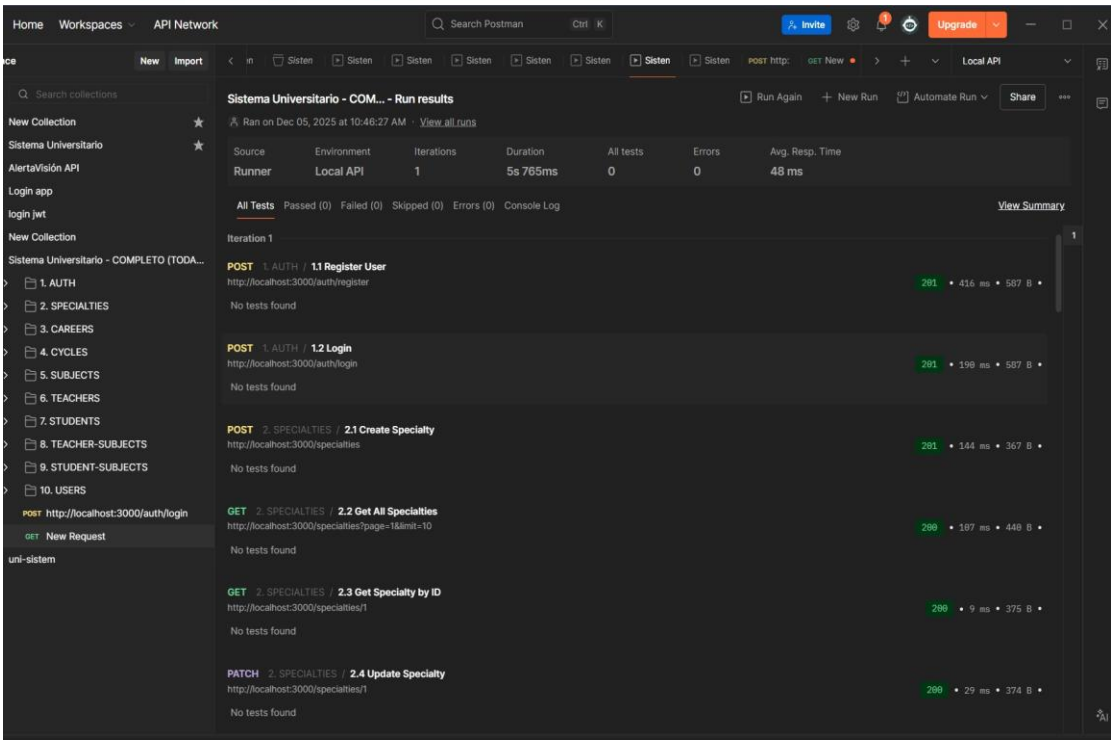
Los principios **ACID** son fundamentales para garantizar el correcto funcionamiento de un sistema universitario, ya que aportan estabilidad, seguridad y confianza en el manejo de la información. Su aplicación permite que el proceso de matriculación se realice de forma controlada, reduciendo errores y asegurando la validez de los datos registrados.

Asimismo, la integración de tecnologías como **NestJS**, **Prisma** y **PostgreSQL** contribuye a implementar estos principios de manera eficiente, permitiendo el desarrollo de una plataforma robusta, organizada y confiable para la gestión académica.

El uso de NestJS, Prisma y PostgreSQL facilita la aplicación de estos principios y permite desarrollar un sistema sólido y bien estructurado.

CAPTURAS DE POSTMAN:

TECNOLÓGICO SUDAMERICANO



All Tests Passed (0) Failed (0) Skipped (0) Errors (0) Console Log View Summa

POST 3. CAREERS / **3.1 Create Career**
http://localhost:3000/careers
201 • 30 ms • 563 B •
No tests found

GET 3. CAREERS / **3.2 Get All Careers**
http://localhost:3000/careers?page=1&limit=10
200 • 7 ms • 623 B •
No tests found

GET 3. CAREERS / **3.3 Get Career by ID**
http://localhost:3000/careers/1
200 • 112 ms • 586 B •
No tests found

PATCH 3. CAREERS / **3.4 Update Career**
http://localhost:3000/careers/1
200 • 39 ms • 570 B •
No tests found

POST 4. CYCLES / **4.1 Create Cycle**
http://localhost:3000/cycles
201 • 9 ms • 360 B •
No tests found

GET 4. CYCLES / **4.2 Get All Cycles**
http://localhost:3000/cycles?page=1&limit=10
200 • 10 ms • 434 B •
No tests found

TECNOLÓGICO SUDAMERICANO

GET 4. CYCLES / **4.3 Get Cycle by ID**
http://localhost:3000/cycles/1
200 • 6 ms • 369 B •
No tests found

PATCH 4. CYCLES / **4.4 Update Cycle**
http://localhost:3000/cycles/1
200 • 18 ms • 367 B •
No tests found

POST 5. SUBJECTS / **5.1 Create Subject**
http://localhost:3000/subjects
201 • 32 ms • 722 B •
No tests found

GET 5. SUBJECTS / **5.2 Get All Subjects**
http://localhost:3000/subjects?page=1&limit=10
200 • 9 ms • 782 B •
No tests found

GET 5. SUBJECTS / **5.3 Get Subject by ID**
http://localhost:3000/subjects/1
200 • 126 ms • 745 B •
No tests found

PATCH 5. SUBJECTS / **5.4 Update Subject**
http://localhost:3000/subjects/1
200 • 45 ms • 729 B •
No tests found

http://localhost:3000/teachers/1
200 • 30 ms • 453 B •
No tests found

POST 7. STUDENTS / **7.1 Create Student**
http://localhost:3000/students
201 • 25 ms • 800 B •
No tests found

GET 7. STUDENTS / **7.2 Get All Students**
http://localhost:3000/students?page=1&limit=10
200 • 10 ms • 860 B •
No tests found

GET 7. STUDENTS / **7.3 Get Student by ID**
http://localhost:3000/students/1
200 • 10 ms • 809 B •
No tests found

PATCH 7. STUDENTS / **7.4 Update Student**
http://localhost:3000/students/1
200 • 39 ms • 807 B •
No tests found

POST 8. TEACHER-SUBJECTS / **8.1 Assign Teacher to Subject**
http://localhost:3000/teacher-subjects
201 • 38 ms • 1.199 KB •
No tests found

TECNOLÓGICO SUDAMERICANO