



UNIVERSITARIA TECNOLÓGICA DE SANTANDER · GRUPO CIAI

UTS Nexus Académico

Necesidad, requerimientos y modelo UML

Qué problema resuelve la plataforma, qué tiene que hacer, qué tiene que cumplir mientras lo hace, y cuatro vistas del sistema: quién lo usa, qué guarda, de qué piezas está hecho y qué ocurre cuando alguien escribe una nota.

SISTEMA	UTS Nexus Académico
INSTITUCIÓN	Universitaria Tecnológica de Santander
CONTENIDO	6 necesidades · 12 requisitos funcionales · 10 no funcionales · 4 diagramas UML
LICENCIA	PolyForm Noncommercial 1.0.0

Contenido

1. **La necesidad** — el problema que se resolvía a mano

2. **Requerimientos**

Funcionales · RF-01 a RF-12

No funcionales · RNF-01 a RNF-10

3. **Modelo UML**

Figura 1 · Casos de uso

Figura 2 · Clases y modelo de datos

Figura 3 · Componentes y despliegue

Figura 4 · Secuencia: registrar una nota

La necesidad

El seguimiento académico de un semestre se llevaba en hojas de cálculo sueltas, cuadernos y correos. Nada de eso está mal hecho: está repartido, y repartido no se puede cruzar a tiempo.

Una planilla por docente

Los pesos del reglamento —30 % trabajos, 60 % parciales, 10 % autoevaluación— se reescriben a mano en cada archivo. Una fórmula mal copiada no da error: da un acta con un número distinto, y se descubre cuando el estudiante reclama.

POR ESO el cálculo tiene que vivir en un solo sitio, fuera de la hoja de cada quien.

La asistencia solo dice sí o no

En papel se marca presente o ausente. Llegar veinte minutos tarde y faltar la clase entera cuentan igual, así que el porcentaje del semestre no representa lo que de verdad pasó en el salón.

POR ESO se registran los minutos reales de cada clase y el porcentaje sale ponderado.

El riesgo se ve cuando ya se perdió

La señal está en los datos desde marzo: notas que bajan entre cortes, faltas seguidas los mismos días. Pero nadie cruza notas con asistencia hasta el cierre, y para entonces ya no hay nada que hacer.

POR ESO el sistema revisa solo y avisa antes, con la razón del aviso escrita.

El consolidado se arma a mano

Coordinación pide los cortes por correo y recibe archivos con formatos distintos. Unificarlos es un trabajo manual que se repite en cada corte y que, además, envejece el mismo día que alguien corrige una nota.

POR ESO el consolidado se genera del mismo dato vivo, no de copias enviadas.

Cada quien ve de más o de menos

Un archivo compartido no distingue roles: o se manda completo, y entonces un docente ve los estudiantes de otro, o se recorta a mano y se pierde el rastro de quién cambió qué.

POR ESO el alcance de cada rol lo decide el servidor y todo cambio queda auditado.

El salón no es el escritorio

Pasar lista ocurre de pie, con el teléfono en la mano y a veces sin wifi; armar el semestre ocurre sentado, con teclado y pantalla grande. Una sola herramienta para las dos cosas sirve mal a las dos.

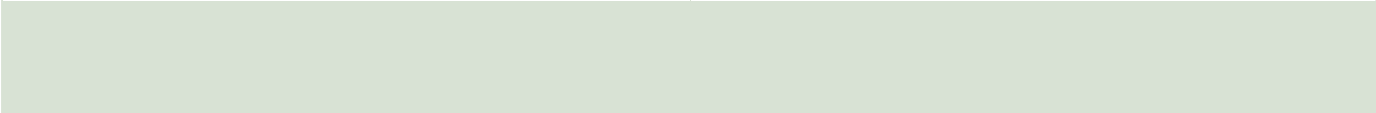
POR ESO hay aplicación de escritorio y de teléfono sobre los mismos datos.

Requerimientos

Lo que el sistema tiene que hacer, y lo que tiene que cumplir mientras lo hace. Cada uno responde a algo de la sección anterior.

Funcionales RF-01 A RF-12 • QUÉ HACE

RF-01 Cuentas, sesión y roles Cinco roles —administración, coordinación, secretaría, docencia y estudiante— con alcance propio. La sesión rota el token en cada canje y reutilizar uno viejo revoca la familia entera.	RF-02 Matrícula y listados de clase El listado se pega como texto, se sube como CSV o se lee de un PDF o una foto. La lectura <i>propone</i> con confianza por fila; escribir es un segundo paso que el docente confirma.
RF-03 Registro de notas por componente Una nota es atómica por estudiante, materia, corte y componente. Se captura a mano, en lote desde una planilla o importada de Excel, PDF o foto.	RF-04 Cálculo del corte y de la definitiva Corte = trabajos 30 % + parciales 60 % + autoevaluación 10 %. Definitiva = C1 33 % + C2 33 % + C3 34 %, se aprueba desde 3.0. A mitad de semestre se promedia solo lo calificado.
RF-05 Asistencia por minutos reales Cada clase registra minutos presentes sobre minutos totales. El porcentaje es ponderado, no un conteo de ausencias.	RF-06 Alerta temprana de riesgo Un modelo cruza notas, tendencia entre cortes y asistencia, y entrega la razón de cada alerta. Si no puede explicarla, no la muestra.
RF-07 Patrones de inasistencia Rachas de faltas seguidas y ausencias repetidas en el mismo día abren un caso de seguimiento. El caso se cierra solo cuando el patrón desaparece; no se borra.	RF-08 Agenda y recordatorios de clase El horario semanal se expande a clases con fecha y hora del campus, junto con parciales, entregas y tutorías. El aviso llega aunque la aplicación esté cerrada.



RF-09 Reportes y consolidados Notas, asistencia y consolidado en PDF y Excel, con membrete, logo y columnas configurables por la institución. Las tres salidas leen el mismo catálogo de columnas.	RF-10 Cierre del periodo académico Abierto, en cierre y cerrado. Durante el cierre se guarda una fotografía por estudiante y materia; reabrir no la borra, anota quién, cuándo y por qué.
RF-11 Sincronización en tiempo real Lo que se cambia en el escritorio aparece en el teléfono sin recargar. El servidor avisa a quien corresponde, no a todo el mundo.	RF-12 Auditoría de cambios Cada modificación guarda autor, fecha y solo lo que cambió, con contraseñas y datos personales enmascarados antes de escribirse.

No funcionales RNF-01 A RNF-10 • CÓMO TIENE QUE HACERLO

RNF-01 Un solo motor de cálculo Ningún cliente recalcula notas, asistencia ni riesgo. Las tres aplicaciones muestran el número que devuelve el servidor, así que no pueden discrepar.	RNF-02 El alcance se decide en el servidor Un docente no ve estudiantes de otro, y coordinación se acota a sus programas. La comprobación es lógica probada, no una pantalla que oculta botones.
RNF-03 Escrituras masivas en un viaje Una planilla de 500 filas por 10 columnas se guarda en una operación por lote, no en quince mil idas y vueltas a la base de datos.	RNF-04 Listados paginados y acotados Todo listado se pide por páginas y la búsqueda ocurre en el servidor: con páginas, filtrar en memoria dejaría fuera a quien no se ha descargado todavía.
RNF-05 Accesibilidad y contraste Contraste AA en los dos modos, foco visible, objetivos táctiles de 48 dp y movimiento reducido cuando el sistema lo pide. El tema tiene tres estados: claro, oscuro y seguir al sistema.	RNF-06 Multiplataforma desde una base común Windows, Linux y Android desde el mismo código de cliente por plataforma. Añadir un sistema operativo no obliga a reescribir la lógica académica.

RNF-07

Degradación declarada

Sin servicio de predicción se usa el motor de reglas y se dice cuál respondió. Sin correo o sin notificaciones el sistema sigue funcionando y lo anota; nada falla en silencio.

RNF-08

Actualización firmada

El instalador verifica la firma antes de aplicar una actualización. Lo que se descarga es un ejecutable: sin firma, cualquiera podría publicar una versión falsa.

RNF-09

Secretos fuera del código

Claves y credenciales viven en el entorno y se validan al arrancar. Las sesiones se guardan en el llavero del sistema operativo, no en un archivo de texto.

RNF-10

Límites de tasa y de tamaño

Cuántas peticiones caben en una ventana y cuánto cabe en una petición son dos defensas distintas, y las dos están puestas. El cupo se cuenta por usuario, no por dirección de red.

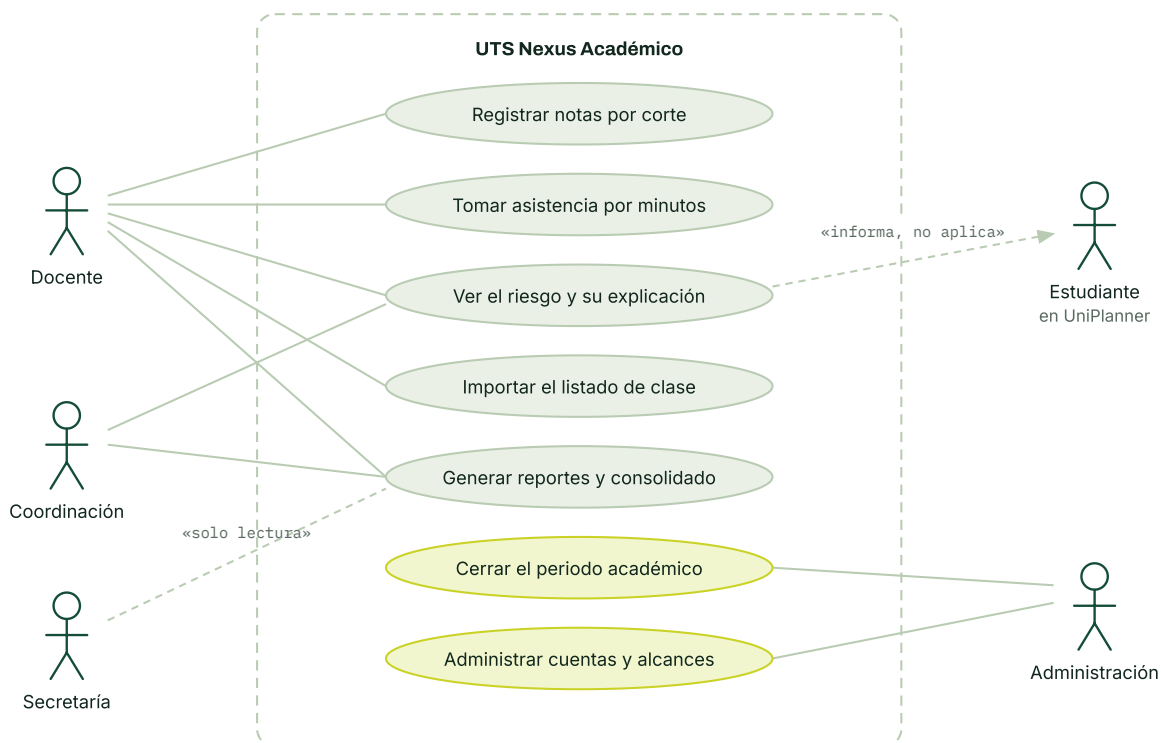
Modelo UML

Cuatro vistas del mismo sistema: quién lo usa, qué guarda, de qué piezas está hecho y qué ocurre cuando alguien escribe una nota.

FIGURA 1

Casos de uso

Diagrama de casos de uso. Quién puede hacer qué. Secretaría hace lo mismo que coordinación pero sin escribir, y el estudiante no entra al sistema: recibe un aviso en su propia aplicación.



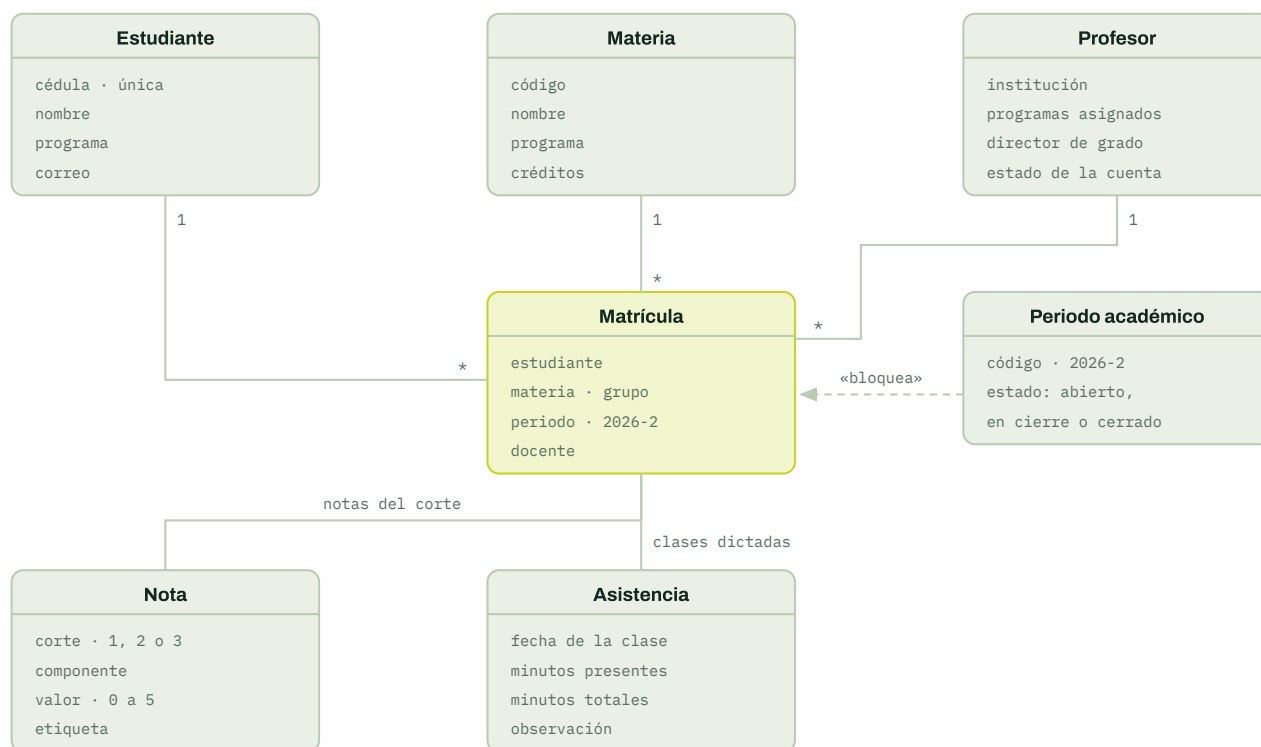
El diagrama, en texto

- **Docente:** registra notas por corte, toma asistencia por minutos, consulta el riesgo con su explicación, importa el listado de clase y genera reportes.
- **Coordinación:** consulta el riesgo y genera reportes y consolidados de sus programas.
- **Secretaría:** los mismos reportes que coordinación, en modo de solo lectura.
- **Administración:** cierra el periodo académico y administra cuentas y alcances. Son los dos casos que ningún otro rol alcanza.
- **Estudiante:** no entra a este sistema. Recibe en UniPlanner un aviso derivado del riesgo, que informa y nunca aplica nada por él.

FIGURA 2

Clases y modelo de datos

Diagrama de clases. El estudiante existe una sola vez en toda la institución, identificado por su cédula. La matrícula es la pieza que lo une a un grupo de una materia en un semestre; todo lo demás cuelga de ahí.



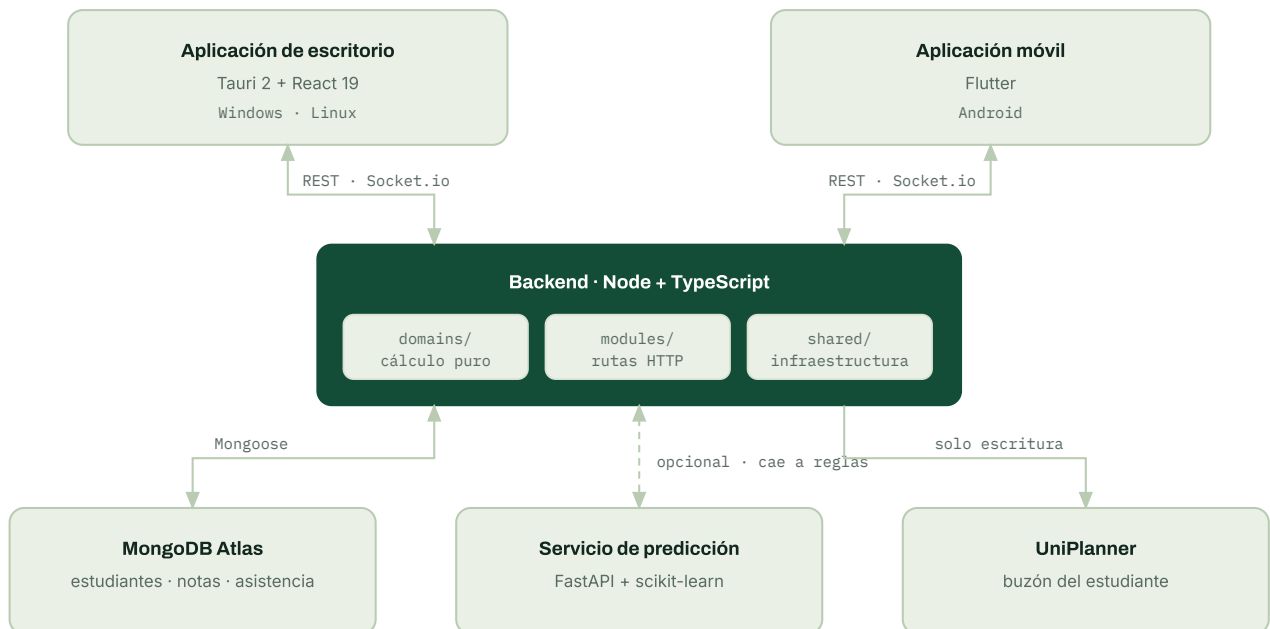
El diagrama, en texto

- **Estudiante** (cédula única, nombre, programa, correo) tiene muchas **matrículas**.
- **Materia** (código, nombre, programa, créditos) tiene muchas matrículas, y **Profesor** (institución, programas asignados, si dirige trabajos de grado) también.
- **Matrícula** une estudiante, materia, grupo, periodo y docente. Es la pieza que define quién ve a quién.
- De la matrícula cuelgan las **notas** —una por corte y componente, con valor de 0 a 5— y las **asistencias**, una por clase, con minutos presentes sobre minutos totales.
- El **periodo académico** bloquea las escrituras de notas, asistencia y matrículas cuando pasa a «en cierre» o «cerrado».

FIGURA 3

Componentes y despliegue

Diagrama de componentes. Un backend y una base de datos en el centro; las aplicaciones solo muestran. El servicio de predicción es opcional: si no responde, el backend usa su motor de reglas y lo dice en la respuesta.



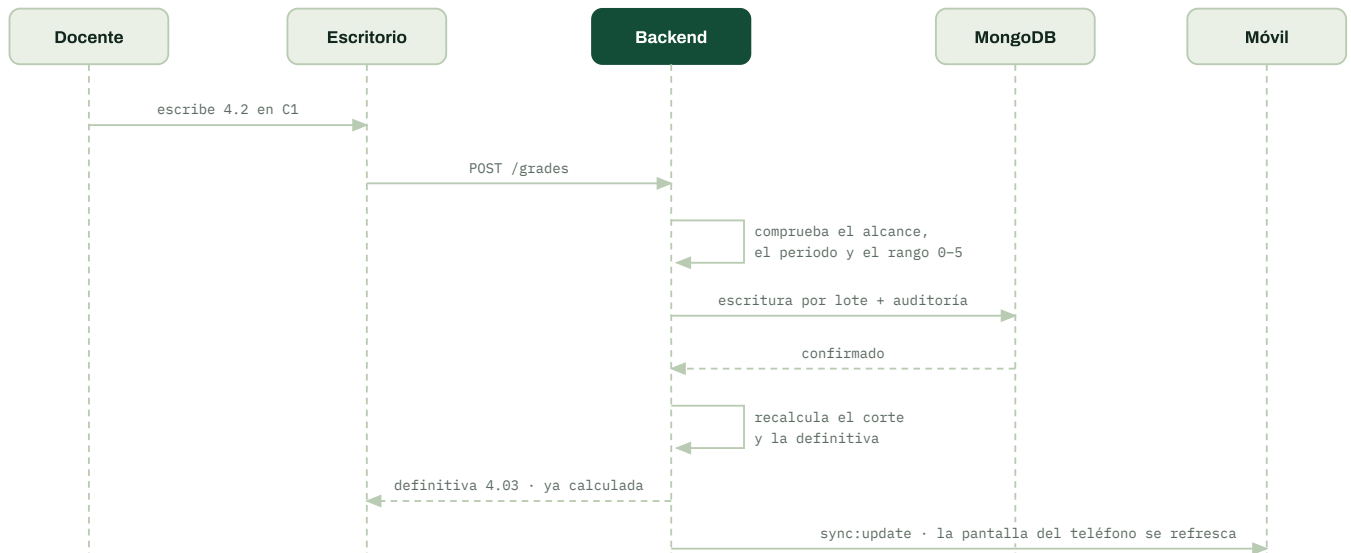
El diagrama, en texto

- Dos aplicaciones cliente: **escritorio** (Tauri 2 + React 19, para Windows y Linux) y **móvil** (Flutter, Android). Las dos hablan con el mismo backend por REST y reciben cambios en vivo por Socket.io.
- El **backend** (Node + TypeScript) tiene tres capas: **domains/** con el cálculo puro y probado, **modules/** con las rutas HTTP por capacidad y **shared/** con la infraestructura común.
- Persiste en **MongoDB Atlas** a través de Mongoose.
- Consulta un **servicio de predicción** en FastAPI. Es opcional: si no responde, el backend usa su motor de reglas y lo declara.
- Escribe avisos en el buzón de **UniPlanner**, la aplicación del estudiante. La flecha va en un solo sentido a propósito: de allí no vuelve ningún dato.

FIGURA 4

Secuencia: registrar una nota

Diagrama de secuencia: registrar una nota. El docente escribe un 4.2; la definitiva la calcula el servidor y vuelve ya hecha. La aplicación no multiplica nada, y por eso el teléfono no puede mostrar otro número.



El diagrama, en texto

1. El docente escribe 4.2 en el primer corte, en la aplicación de escritorio.
2. El escritorio envía la nota al backend con una petición POST /grades.
3. El backend comprueba que ese estudiante está en su alcance, que el periodo sigue abierto y que la nota está entre 0 y 5.
4. Guarda en MongoDB con una escritura por lote y deja el registro de auditoría en la misma operación.
5. Recalcula el corte y la definitiva con los pesos del reglamento.
6. Devuelve al escritorio la definitiva ya calculada: la aplicación no multiplica nada.
7. Emite un evento a los dispositivos de ese usuario, y la pantalla del teléfono se refresca sola.