|  |
| --- |
|  |
| Proyecto: Alcanzando la nota  Documento de Arquitectura de Software |
| Version 1.0 |
| Compañia DJG Tech  **Integrantes:** Dillan Alexander Asprilla  Juan Manuel Conde Aldana  Gustavo Adolfo Camargo Pineda |
|  |
| **Fecha 21/05/2025** |

**Historia de Revisiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Fecha** | **Autor** | **Revisor** | **Cambios** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Tabla de Contenido**

[1. Descripción del Documento 5](#_Toc198742817)

[1.1. Propósito 5](#_Toc198742818)

[1.2. Audiencia 5](#_Toc198742819)

[1.3. Document Organization 6](#_Toc198742820)

[1.4. Terminología y Definiciones 7](#_Toc198742821)

[1.5. Documentación del Proyecto Relacionada. 7](#_Toc198742822)

[1.6. Referencias Externas 7](#_Toc198742823)

[2. Visión General del Proyecto 7](#_Toc198742824)

[2.1. Contexto del negocio 7](#_Toc198742825)

[2.2. Contexto del Sistema 8](#_Toc198742826)

[2.3. Participantes Claves 9](#_Toc198742827)

[2.4. Interesados del Proyecto 10](#_Toc198742828)

[3. Drivers Arquitectónicos 12](#_Toc198742829)

[3.1. Restricciones de Negocio 12](#_Toc198742830)

[3.2. Restricciones Técnicas. 12](#_Toc198742831)

[3.3. Requisitos Funcionales Claves. 12](#_Toc198742832)

[3.4. Requisitos de Atributos de Calidad 13](#_Toc198742833)

[3.5. Interesados vs Requisitos No Funcionales. 14](#_Toc198742834)

[4. Tácticas de Arquitectura y Estilos Arquitectónicos. 15](#_Toc198742835)

[4.1. Tácticas 15](#_Toc198742836)

[Tácticas de detección y recuperación de fallos: 15](#_Toc198742837)

[Tácticas de autenticación y autorización: 15](#_Toc198742838)

[Tácticas de protección de datos: 16](#_Toc198742839)

[Tácticas de gestión de recursos: 17](#_Toc198742840)

[Tácticas de eficiencia en procesamiento: 17](#_Toc198742841)

[Tácticas de escalado horizontal: 17](#_Toc198742842)

[Tácticas de distribución de carga: 17](#_Toc198742843)

[Tácticas de modularidad: 18](#_Toc198742844)

[Tácticas de despliegue y mantenimiento: 18](#_Toc198742845)

[4.2. Patrones Arquitectonicos 18](#_Toc198742846)

[4.3. Estrategia Arquitectónica Global 21](#_Toc198742847)

[5. Vistas de Arquitectura 23](#_Toc198742848)

[5.1. Escenarios Arquitectónicos Identificados 23](#_Toc198742849)

[5.2. Vista Lógica 25](#_Toc198742850)

[5.3. Vista de Implementación 27](#_Toc198742851)

[5.4. Vista de Procesos 29](#_Toc198742852)

[5.5. Vista Física 31](#_Toc198742853)

[6. Directorio 35](#_Toc198742854)

[6.1. Glosario 35](#_Toc198742855)

[6.2. Acrónimos 36](#_Toc198742856)

[7. Apéndice 39](#_Toc198742857)

[7.1. Lista de Pruebas de Concepto 39](#_Toc198742858)

# Descripción del Documento

## Propósito

El propósito de este documento es establecer un consenso en el diseño de la arquitectura de software del sistema "Alcanzando la Nota" entre los miembros del equipo de trabajo, stakeholders del Ministerio de Educación y demás partes interesadas. Este documento servirá como guía definitiva durante la implementación, mantenimiento y evolución del sistema, documentando las decisiones técnicas, trade-offs considerados y la justificación de la arquitectura seleccionada.

Este documento de arquitectura detalla la estructura de alto nivel del sistema nacional de evaluación, incluyendo componentes, interacciones, vistas y patrones arquitectónicos. Define cómo se abordarán los requisitos funcionales y no funcionales críticos identificados, especialmente disponibilidad, seguridad, rendimiento, escalabilidad y mantenibilidad. Además, proporciona la base para validar que la implementación técnica satisfaga las necesidades del negocio y cumpla con las restricciones operativas y presupuestarias del Ministerio de Educación.

La arquitectura documentada aquí representa un diseño serverless en AWS que permitirá gestionar eficientemente la aplicación de pruebas nacionales para aproximadamente 40,000 estudiantes, 2,000 calificadores y 50 administradores, garantizando un SLA de 99.9% de disponibilidad durante las ventanas de evaluación y un estricto cumplimiento de los requisitos de seguridad y auditabilidad.

# Visión General del Proyecto

## Contexto del negocio

El **Ministerio de Educación** impulsa la creación de un **sistema nacional unificado** para la aplicación y calificación de pruebas estandarizadas en todos los colegios del país. El objetivo es garantizar **equidad, transparencia y trazabilidad** en los procesos de evaluación, permitiendo comparar resultados entre instituciones, profesores y estudiantes de manera objetiva y consistente.

El sistema deberá dar servicio a:

* **≈ 40 000 estudiantes**, quienes solo podrán presentar las pruebas dentro de los centros de evaluación oficialmente designados, bajo condiciones controladas que preserven la integridad del examen.
* **≈ 2 000 calificadores**, responsables de ingresar manualmente las notas de las preguntas de respuesta corta y ensayos.
* **≈ 50 administradores**, encargados de la gestión operativa y el soporte del sistema.

Las pruebas incluirán preguntas de **selección múltiple, respuesta corta y ensayos**. El sistema deberá consolidar los resultados nacionales en un repositorio único que permita emitir reportes detallados sobre qué estudiantes presentaron la prueba y qué puntaje obtuvieron.

Cualquier modificación a las calificaciones requerirá la aprobación de **tres entidades gubernamentales**, asegurando un estricto control de calidad y legitimidad de los resultados. Además, el proyecto debe gestionar su presupuesto de manera justificable cada año fiscal. Dado que el país **no cuenta con infraestructura propia de hosting**, se recurre a un proveedor externo; por ello, el modelo de costos operativos deberá ser transparente, escalable y sujeto a revisión periódica.

## Participantes Claves

* + 1. **Arquitecto de Software**

El Arquitecto de Software es responsable del diseño técnico del sistema "Alcanzando la Nota", definiendo la estructura serverless en AWS y los patrones arquitectónicos que cumplirán con los requisitos funcionales y no funcionales. Establece las estrategias técnicas para los tres modos de ejecución de exámenes, supervisa el diseño del flujo de aprobación de notas y define las prácticas de IaC y CI/CD. Evalúa y mitiga riesgos técnicos relacionados con disponibilidad y seguridad.

* + 1. **Gerente de Proyecto**

El Gerente de Proyecto coordina la planificación, ejecución y monitoreo de todas las actividades del proyecto, actuando como enlace principal entre DJG Tech y el Ministerio de Educación. Gestiona el cronograma, presupuesto y recursos, asegurando el cumplimiento de plazos y la calidad del entregable. Es responsable de identificar riesgos, implementar medidas de mitigación y facilitar la comunicación efectiva entre todos los interesados.

* + 1. **Líder de Desarrollo**

El Líder de Desarrollo supervisa la implementación técnica del sistema según las especificaciones arquitectónicas. Dirige al equipo de desarrolladores, establece estándares de codificación, realiza revisiones de código y garantiza la calidad técnica de la solución. Trabaja estrechamente con el Arquitecto para materializar el diseño y resolver desafíos técnicos durante la implementación de los microservicios y componentes event-driven.

* + 1. **Especialista en DevOps/SRE**

El Especialista en DevOps/SRE implementa la infraestructura como código, configura los pipelines de CI/CD y establece los sistemas de monitoreo para garantizar el SLA de 99.9% durante las ventanas de evaluación. Es responsable de las estrategias de despliegue canary, las configuraciones de auto-scaling y la implementación de las tácticas de disponibilidad como réplicas multi-AZ y health-checks.

* + 1. **Especialista en Seguridad**

El Especialista en Seguridad diseña e implementa los controles de seguridad del sistema, incluyendo el cifrado en tránsito/reposo con KMS, la configuración de IAM con roles granulares, la autenticación MFA y los mecanismos de attestation para dispositivos cliente. Realiza evaluaciones de vulnerabilidades, supervisa el cumplimiento normativo y asegura la protección de los datos sensibles de estudiantes y calificaciones.

* + 1. **Analista de Negocios/Requisitos**

El Analista de Negocios actúa como puente entre los requisitos educativos del Ministerio y las especificaciones técnicas. Recopila, documenta y valida los requerimientos funcionales y no funcionales, asegurando que la solución responda a las necesidades reales de los calificadores, estudiantes y administradores. Elabora historias de usuario detalladas y criterios de aceptación para los distintos componentes del sistema.

* + 1. **Representante del Ministerio de Educación**

El Representante del Ministerio aporta el conocimiento del dominio educativo, define las políticas de evaluación y valida que el sistema cumpla con los requisitos pedagógicos y administrativos establecidos. Participa en la toma de decisiones críticas, especialmente en aspectos relacionados con el flujo de aprobación de notas, la generación de reportes y la interpretación de resultados nacionales.

* + 1. **Especialista en UX/UI**

El Especialista en UX/UI diseña interfaces intuitivas y accesibles para los tres perfiles de usuario (estudiantes, calificadores y administradores). Adapta la experiencia para los diferentes modos de ejecución de exámenes, asegurando consistencia y facilidad de uso independientemente del dispositivo o modalidad. Realiza pruebas de usabilidad y optimiza la interfaz para cumplir con el requisito de latencia menor a 300ms.

## Interesados del Proyecto

El sistema "Alcanzando la Nota" debe satisfacer las necesidades de diversos interesados, cada uno con expectativas específicas y prioridades diferentes:

* + 1. **Ministerio de Educación**

Principal stakeholder y patrocinador del proyecto. Sus intereses principales son:

* Regulación y estandarización de las evaluaciones a nivel nacional
* Transparencia y auditabilidad en el proceso de calificación
* Optimización del presupuesto y justificación de costos anuales
* Obtención de métricas educativas confiables para toma de decisiones
  + 1. **Estudiantes**

Usuarios finales primarios del sistema con aproximadamente 40,000 individuos. Sus intereses incluyen:

* Experiencia de usuario intuitiva y accesible
* Rendimiento adecuado sin latencias excesivas durante los exámenes
* Fiabilidad del sistema para evitar pérdida de respuestas
* Seguridad de sus datos personales y calificaciones
  + 1. **Calificadores**

Grupo de aproximadamente 2,000 profesionales responsables de evaluar exámenes. Sus intereses son:

* Herramientas eficientes para la calificación de respuestas abiertas
* Seguridad y trazabilidad en el proceso de asignación de notas
* Interfaces productivas que optimicen su trabajo
* Confiabilidad en el almacenamiento y procesamiento de calificaciones
  + 1. **Administradores**

Grupo de aproximadamente 50 personas encargadas de gestionar el sistema. Sus intereses incluyen:

* Herramientas efectivas para operaciones y monitoreo
* Capacidad para generar reportes detallados
* Funcionalidades de administración de usuarios y sedes
* Mantenibilidad del sistema a largo plazo
  + 1. **Equipo TI/DevOps**

Personal técnico responsable del mantenimiento y operación del sistema. Sus intereses son:

* Infraestructura como código para facilitar despliegues
* Sistemas de monitoreo y alertas efectivos
* Capacidad de escalar recursos según demanda
* Seguridad de la infraestructura y los datos
  + 1. **Proveedor de hosting (AWS)**

Socio tecnológico que proporciona la infraestructura cloud. Sus intereses incluyen:

* Cumplimiento de los SLAs establecidos
* Optimización de costos y recursos
* Implementación siguiendo buenas prácticas cloud
* Seguridad de la plataforma

# Requerimientos

## Requerimientos Funcionales (RF)

|  |  |  |
| --- | --- | --- |
| ID | Descripción | Observaciones Críticas / Posibles Riesgos |
| RF-01 | **Gestión de usuarios**: Alta, baja y actualización de estudiantes, calificadores y administradores, con perfiles y permisos diferenciados. | Asume que todos los datos personales pueden ser gestionados en línea; podría requerirse sincronización con registros ministeriales externos. |
| RF-02 | **Planificación de convocatorias**: Creación de periodos de evaluación, asignación de fechas y centros autorizados, capacidad máxima por sede y turno. | ¿Quién valida la capacidad real de los centros? Puede impactar la logística y la experiencia de usuario. |
| RF-03 | **Creación y mantenimiento de bancos de preguntas**: Soporte para selección múltiple, respuesta corta y ensayo. | La calidad del banco afecta directamente la validez psicométrica de la prueba; podría requerir flujos de revisión pedagógica fuera del sistema. |
| RF-04 | **Generación de exámenes**: Selección aleatoria o definida de preguntas por zona/temario, con versiones paralelas para minimizar fraude. | Supone que la aleatorización cumple estándares de equivalencia entre versiones. |
| RF-05 | **Distribución segura de exámenes**: Descarga/streaming controlado en dispositivos del centro o propios (según modalidad), con temporizador y bloqueo de navegación. | El “modo kiosk” en dispositivos propios es complejo: se necesitarían agentes o perfiles MDM. |
| RF-06 | **Ejecución y auto-guardado de respuestas**: Guardado incremental para prevenir pérdida de datos por fallos de conexión. | Requiere tolerancia a redes internas inestables en zonas rurales. |
| RF-07 | **Entrega automática y cierre**: Finalización de la sesión cuando expire el tiempo, con firma digital de integridad del intento. | Debe discutirse qué ocurre ante cortes de energía prolongados. |
| RF-08 | **Calificación automática** para selección múltiple y cálculo de puntajes preliminares. | Validar sesgos de puntuación en preguntas complejas (p. ej., multiponderación). |
| RF-09 | **Calificación manual**: Interfaz para que calificadores asignen notas a respuestas cortas/ensayos, con criterios y rúbricas visibles. | Posible cuello de botella si la carga de ensayos es alta. |
| RF-10 | **Workflow de aprobación de cambios de nota**: Flujo secuencial con firma digital de tres entidades gubernamentales y trazabilidad completa. | Puede alargar significativamente los plazos de publicación de resultados. |
| RF-11 | **Consolidación y publicación de resultados**: Agregación nacional por estudiante, profesor, colegio y otras dimensiones analíticas. | Implica definir claves maestras de institución y docente para evitar colisiones. |
| RF-12 | **Reportes y exportación**: Generación de reportes CSV/PDF con filtros por rol, sede, rango de fechas y estado de calificación. | El volumen de datos históricos podría requerir partición o archivado. |
| RF-13 | **Auditoría y bitácora**: Registro inmutable de accesos, cambios de configuración y eventos críticos. | Se debe acordar el período de retención legal de logs. |
| RF-14 | **Notificaciones**: Alertas por email/SMS a estudiantes (confirmación de inscripción), calificadores (tareas pendientes) y administradores (incidentes). | Debe respetar normativas de datos personales (consentimiento de estudiantes menores). |
| RF-15 | **Autoservicio de soporte**: FAQs, ticketing y trazabilidad de incidentes. | Podría reducir carga de mesa de ayuda en picos de uso. |

## Requerimientos No Funcionales (RNF)

|  |  |  |  |
| --- | --- | --- | --- |
| Categoría | ID | Descripción | Métrica / Criterio |
| **Disponibilidad** | RNF-A1 | El sistema estará operativo al menos el 99.9 % del tiempo durante las ventanas de examen. | ≤ 44 min de inactividad al mes en periodos críticos. |
| **Rendimiento** | RNF-P1 | Tiempo de respuesta máximo de 300 ms (p95) con hasta 40 000 usuarios concurrentes. | Medido desde la solicitud del cliente hasta la respuesta del servicio de backend. |
| **Escalabilidad** | RNF-S1 | El sistema escalará horizontalmente (x 10) sin intervención manual durante aumentos de carga programados. | De 4 000 a 40 000 usuarios en ≤ 5 min sin degradación. |
| **Seguridad** | RNF-SEC1 | Cifrado de datos en tránsito y reposo con algoritmos aprobados por el Ministerio (p. ej., AES-256, TLS 1.3). | 100 % del tráfico HTTPS; llaves rotadas anualmente o ante incidentes. |
|  | RNF-SEC2 | Autenticación multifactor (MFA) requerida para roles privilegiados. | 100 % de administradores y calificadores. |
|  | RNF-SEC3 | Cumplimiento de la Ley nacional de protección de datos y FERPA-like (si aplica). | Auditorías anuales sin hallazgos críticos. |
| **Usabilidad** | RNF-U1 | Curva de aprendizaje < 15 min para estudiantes (encuesta SUS ≥ 80). | Pruebas piloto con ≥ 50 estudiantes. |
| **Mantenibilidad** | RNF-M1 | Cambios menores desplegados a producción en ≤ 1 h con baja probabilidad de regresión (< 5 %). | Pipeline CI/CD automatizado con pruebas ≥ 80 % coverage. |
| **Portabilidad** | RNF-PORT1 | Cliente web debe funcionar en navegadores con cuotas de mercado ≥ 5 % y en sistemas operativos Windows, macOS, Linux. | Compatibilidad verificada en cada release. |
| **Accesibilidad** | RNF-ACC1 | Cumplimiento WCAG 2.1 nivel AA para interfaces de estudiante y calificador. | Auditorías semestrales de accesibilidad. |
| **Interoperabilidad** | RNF-INT1 | APIs expuestas REST/GraphQL documentadas con OpenAPI, permitiendo integración con sistemas de analítica externos. | Latencia adicional < 50 ms para llamadas internas. |
| **Auditabilidad** | RNF-AUD1 | Toda acción que altere datos de calificación debe almacenarse con sello de tiempo, identificador de usuario y hash de integridad. | 100 % de eventos críticos registrados. |
| **Fiabilidad** | RNF-R1 | Pérdida de datos por fallo catastrófico ≤ 0.01 % gracias a políticas de respaldo y DR. | RPO 5 min, RTO 30 min. |
| **Costo-Eficiencia** | RNF-COST1 | El costo operativo anual no debe superar el presupuesto aprobado (+10 % margen). | Informes trimestrales de coste/usuario. |
| **Legal / Compliance** | RNF-L1 | Mecanismos de consentimiento para menores y padres, conforme a regulaciones locales. | Formularios electrónicos almacenados y verificables. |

# Atributos de Calidad

|  |  |  |
| --- | --- | --- |
| Atributo | Razonamiento (Qué protege / habilita) | Escenario Ejemplo (Who, When, What Stimulus) |
| **Disponibilidad** | Asegura que la ventana de examen no se vea interrumpida, protegiendo la confiabilidad del proceso educativo nacional. | *Durante un examen nacional, un AZ falla; el sistema debe redireccionar tráfico sin que los estudiantes noten la caída.* |
| **Rendimiento** | Evita frustración y tiempos muertos en centros con muchos estudiantes conectados simultáneamente. | *Estudiante envía la respuesta y recibe confirmación en < 300 ms pese a 40 k usuarios activos.* |
| **Escalabilidad** | Permite afrontar picos estacionales sin sobredimensionar la infraestructura todo el año. | *Tráfico sube 10× en 5 min; la plataforma añade instancias automáticamente y mantiene SLA.* |
| **Seguridad** | Protege datos sensibles (resultados, PII) y mantiene la legitimidad del examen. | *Admin intenta acceder a resultados fuera de su región; el sistema bloquea y lo registra.* |
| **Usabilidad** | Reduce errores de los estudiantes y la curva de adopción de calificadores; minimiza costos de soporte. | *Estudiante con baja alfabetización digital completa el examen sin asistencia.* |
| **Mantenibilidad** | Facilita adaptación a futuros cambios curriculares o legislativos sin reescribir toda la plataforma. | *Nuevo tipo de pregunta debe añadirse en un sprint sin riesgos altos.* |
| **Auditabilidad** | Ofrece evidencias inmutables ante disputas o solicitudes de transparencia pública. | *Padre apela la calificación; auditor extrae el historial completo del intento.* |
| **Reusabilidad / Modularidad** | Componentes como “Gestión de Usuarios” o “Reportes” podrían usarse en otros proyectos ministeriales. | *Otro organismo usa el módulo de identidad sin modificar código fuente.* |
| **Portabilidad** | Evita lock-in excesivo y facilita migraciones futuras o despliegues en nubes soberanas. | *Cambio de proveedor cloud o despliegue en región local sin reescritura significativa.* |
| **Accesibilidad** | Asegura el derecho a la educación de estudiantes con discapacidad. | *Usuario con lector de pantalla completa la prueba con éxito.* |
| **Confiabilidad** | Minimiza la probabilidad de resultados perdidos o inconsistentes. | *Error de red interrumpe sesión; sistema continúa desde última pregunta guardada.* |
| **Observabilidad** | Permite detectar y diagnosticar fallos antes de que afecten la experiencia de examen. | *Dashboards alertan a la mesa de ayuda cuando la latencia supera 200 ms.* |
| **Costo-Eficiencia** | Mantiene la viabilidad del proyecto en ciclos presupuestales anuales. | *Análisis de costo por estudiante muestra tendencia decreciente año contra año.* |

# Drivers Arquitectónicos

## Matriz Stakeholder x Atributos (ponderación interna de cada interesado)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stakeholder / Attribute | Liability | Security | Performance | Maintainability | Compatibility | Usability | Availability | **Total** |
| **Ministerio de Educación** | 20 | 20 | 10 | 10 | 10 | 10 | 20 | **100** |
| **Estudiantes** | 5 | 10 | 20 | 0 | 15 | 30 | 20 | **100** |
| **Calificadores** | 10 | 15 | 15 | 5 | 10 | 25 | 20 | **100** |
| **Administradores** | 10 | 15 | 10 | 25 | 10 | 10 | 20 | **100** |
| **Equipo TI / DevOps** | 5 | 20 | 20 | 25 | 10 | 5 | 15 | **100** |
| **Proveedor AWS** | 5 | 15 | 20 | 10 | 10 | 5 | 35 | **100** |
| **Attribute Total (%)** | **9 %** | **15.8 %** | **15.8 %** | **12.5 %** | **10.8 %** | **14.2 %** | **21.7 %** | **100 %** |

## Priorización de atributos de calidad

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Description (resumida) | Representative Metric(s) | Impact (1-3) | Difficulty (1-3) | **Weight** (priority %) | **Value** (I+D) × W |
| **Availability** | Capacidad del sistema para estar operativo durante las ventanas de examen. | Uptime ≥ 99.9 %; RTO ≤ 30 min | 3 | 3 | **25** | **150** |
| **Security** | Protección de datos personales y resultados. | % datos cifrados; nº vulnerabilidades críticas | 3 | 3 | **15** | **90** |
| **Performance** | Tiempo de respuesta y throughput bajo 40 000 concurrencias. | p95 ≤ 300 ms; req/s | 3 | 2 | **15** | **75** |
| **Usability** | Facilidad de uso para estudiantes y calificadores. | SUS ≥ 80; % tareas completadas | 3 | 2 | **15** | **75** |
| **Maintainability** | Facilidad de evolutivo, depuración y despliegues. | MTTR < 3 días; cobertura 80 % | 2 | 2 | **12** | **48** |
| **Compatibility** | Integración con otros sistemas y dispositivos heterogéneos. | Tests de interoperabilidad; compat. navegadores | 2 | 2 | **10** | **40** |
| **Liability** | Trazabilidad y responsabilidades legales ante disputas. | Nº incidentes legales; score de compliance | 2 | 3 | **8** | **40** |
| **Σ Weights = 100 %** |  |  |  |  | **—** | **518** |

## Atributos de Calidad Seleccionados

Los siguientes cinco atributos de calidad han sido identificados como determinantes para la toma de decisiones de diseño y construcción del sistema **“Alcanzando la Nota”**. Su selección deriva de los requerimientos de negocio, las restricciones regulatorias y el análisis de riesgo asociado al proyecto. Cada apartado describe la motivación estratégica, un escenario de evaluación representativo y las implicaciones directas sobre la arquitectura.

* + 1. **Disponibilidad**

La continuidad del servicio durante las ventanas oficiales de examen constituye un requisito crítico de negocio. Una interrupción comprometería la validez de la prueba a nivel nacional y generaría costos de reprogramación inaceptables.

**Escenario de referencia**

*“Durante una sesión oficial, un centro de datos sufre una caída completa. El sistema mantiene una disponibilidad ≥ 99.9 % y la latencia p95 no se incrementa en más de 50 ms.”*

Implicaciones arquitectónicas:

* Implementación en configuración activo/activo entre zonas de disponibilidad independientes.
* Diseño sin estado en la capa de cómputo y orquestación automática de fail-over para componentes con estado.
* Ensayos de resiliencia programados (GameDay) antes de cada periodo de evaluación.
  + 1. **Seguridad**

El sistema procesa información sensible de menores y resultados académicos oficiales; la confidencialidad, integridad y trazabilidad de estos datos son obligaciones legales y éticas ineludibles.

**Escenario de referencia**

*“Se intenta modificar una calificación consolidada mediante credenciales comprometidas. La operación es bloqueada, el evento queda registrado en un almacén inmutable y la entidad auditora dispone de evidencia forense completa.”*

Implicaciones arquitectónicas:

* Cifrado de extremo a extremo (TLS 1.3 y claves gestionadas en HSM/KMS) y autenticación multifactor para funciones privilegiadas.
* Registro inmutable de eventos mediante un patrón de *ledger* o *event sourcing* que permita no repudio.
* Segmentación de red y política de mínimos privilegios para todas las identidades de servicio y usuario.
  + 1. **Rendimiento**

Las evaluaciones nacionales concentran hasta 40 000 usuarios concurrentes en periodos extremadamente breves. El sistema debe absorber estos picos sin degradar la experiencia.

**Escenario de referencia**

*“La concurrencia se multiplica por diez en menos de cinco minutos. El sistema ajusta capacidad de forma automática y mantiene un tiempo de respuesta ≤ 300 ms en el percentil 95.”*

Implicaciones arquitectónicas:

* Separación estricta entre presentación, lógica de negocio y procesamiento asíncrono mediante colas o eventos.
* Estrategias de caché en borde y en memoria (CDN, DAX/ElastiCache) para contenido estático y consultas de lectura frecuente.
* Escalado programado y reservas de concurrencia previa a la franja de examen a fin de contener los costos.
  + 1. **Usabilidad**

El público objetivo incluye estudiantes de diversas regiones y capacidades; la plataforma debe ofrecer una interfaz intuitiva, inclusiva y robusta frente a conexiones inestables.

**Escenario de referencia**

*“Un estudiante con una conexión de 2 Mbps en un dispositivo de gama media completa la prueba sin asistencia; un usuario con lector de pantalla recorre todo el flujo cumpliendo WCAG 2.1 AA.”*

Implicaciones arquitectónicas:

* Aplicación web progresiva ligera con capacidad de trabajo desconectado y sincronización diferida.
* Diseño *responsive* y validaciones automáticas de accesibilidad integradas en el proceso CI/CD.
* Guardado automático de respuestas y temporización clara para minimizar la ansiedad del usuario.
  + 1. **Mantenibilidad**

La normativa educativa y los contenidos curriculares cambian con frecuencia. El sistema debe permitir incorporaciones funcionales ágiles sin poner en riesgo la estabilidad operativa.

**Escenario de referencia**

*“Se introduce un nuevo tipo de pregunta interactiva y se despliega en producción en menos de diez días, sin afectar los niveles de servicio acordados.”*

Implicaciones arquitectónicas

* Descomposición en dominios independientes con contratos de API bien definidos y versionados.
* Infraestructura como código, pipelines de integración continua con *feature flags* y despliegues canary.
* Instrumentación de observabilidad integral (tracing distribuido, métricas y registros estructurados) para facilitar el diagnóstico y la mejora continua.

# Tácticas de Arquitectura y Estilos Arquitectónicos.

## Tácticas

* + 1. **Disponibilidad**

### ****Tácticas para Detección Proactiva****

* **Circuit Breaker distribuido** Biblioteca de resiliencia en cada microservicio; tras N fallos abre el circuito 60 s, evitando la cascada.en modo Multi-AZ para garantizar la disponibilidad de datos.
* **Health-checks sintéticos multi-AZ** Probes externos emulan flujos de usuario cada 60 s; errores consecutivos disparan alarmas de alta prioridad.

### ****Tácticas para Recuperación Automática****

* **Configuración activo/activo** Aurora Serverless v2, DynamoDB global tables y EventBridge replicados en dos zonas; fail-over DNS < 30 s.
* **Reintentos con back-off exponencial + Jitter** Aplicados en SDK AWS y colas SQS para absorber fallos transitorios.

### ****Tácticas para Resiliencia Verificada****

* **GameDay trimestral** Inyección de fallos (SsmChaos, Fault Injection Simulator) con métricas de éxito pactadas: RTO ≤ 30 min, Error rate ≤ 0.1 %.
  + 1. **Seguridad**

### ****Tácticas para Autenticación y Autorización****

* **Cognito + MFA adaptativa** TOTP obligatorio para calificadores/administradores; estudiantes usan contraseña fuerte y verificación de centro/IP.
* **RBAC granular en IAM** Políticas mínimas; separación de duties entre operación y desarrollo (principio PoLP).

### ****Tácticas para Protección de Datos****

* **Cifrado “defence-in-depth”** TLS 1.3 en tránsito; AES-256 en reposo con KMS keys separadas por dominio de datos.
* **Ledger Database para calificaciones** Registro inmutable (Q LDB o DynamoDB + AWS Audit Manager) garante de no repudio.
* **Tokenización PII** Identificadores sensibles reemplazados por UUID; tabla de look-up cifrada con acceso restringido al módulo “Identity”.
  + 1. **Rendimiento**

### ****Tácticas para Gestión Eficiente de Recursos****

* **Caché multinivel** CloudFront para assets; API Gateway caching; DynamoDB DAX para lecturas “hot”; ElastiCache Redis para métricas de examen.
* **Persistencia optimizada** Particionamiento por *ExamId* + StudentId; índices proyectados minimizan *hot partitions*.
* **Procesamiento asíncrono** Preguntas de ensayo y reporting en colas   
  SQS → Lambda → Step Functions, aislando picos de carga.

### ****Tácticas de Elasticidad Controlada****

* **Provisioned Concurrency programado** Lambdas del flujo de examen precalentadas 15 min antes del inicio oficial.
* **Auto-scaling basado en métricas** Target tracking para throughput API y consumo de stream DynamoDB; escala lineal 1:2500 usuarios.
  + 1. **Usabilidad**
* **PWA ligera** Resource hinting, Service Worker y precache; permite modo offline de hasta 10 min con reconciliación posterior.**Auto-scaling basado en métricas** Target tracking para throughput API y consumo de stream DynamoDB; escala lineal 1:2500 usuarios.
* **Diseño accesible (WCAG 2.1 AA)** Contraste, navegación por teclado, roles ARIA y lectura de pantalla comprobados en CI con Pa11y.
* **Guardado automático incremental** Draft cada 30 s o al cambiar de pregunta; feedback visual de “respuesta guardada”.
* **Carga progresiva** Lazy loading de imágenes/recursos; prioridad a elementos above-the-fold.
* **Indicadores de tiempo y conectividad** Barra persistente muestra tiempo restante y estado de conexión; reduce ansiedad y llamadas de soporte.
  + 1. **Mantenibilidad**

### ****Tácticas para Modularidad y Encapsulamiento****

* **Microservicios orientados a dominio** Exam, Submission, Grading, Approval, Reporting; contratos REST/GraphQL versionados.
* **Separación de capas** API → Orquestación (Lambda/Step Functions) → Reglas de dominio → Persistencia; facilita pruebas unitarias.
* **Procesamiento asíncrono** Preguntas de ensayo y reporting en colas   
  SQS → Lambda → Step Functions, aislando picos de carga.

### ****Tácticas de Elasticidad Controlada****

* **Provisioned Concurrency programado** Lambdas del flujo de examen precalentadas 15 min antes del inicio oficial.
* **Auto-scaling basado en métricas** Target tracking para throughput API y consumo de stream DynamoDB; escala lineal 1:2500 usuarios.

## Patrones Arquitectonicos

La solución “Alcanzando la Nota” incorporará los siguientes **patrones arquitectónicos**; su selección se fundamenta en los cinco drivers definidos (Disponibilidad, Seguridad, Rendimiento/Escalabilidad, Usabilidad/Accesibilidad y Mantenibilidad) y en las restricciones de negocio y operación del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Nº | Patrón | Aplicación en el sistema | Objetivos de calidad reforzados |
| **P-1** | **Microservicios orientados a dominio (DDD + Bounded Contexts)** | Los dominios Exam, Submission, Grading, Approval y Reporting se implementan como servicios independientes, con contratos de API claros. Cada dominio escala, se actualiza y se despliega de forma autónoma. | Rendimiento / Escalabilidad, Mantenibilidad, Disponibilidad |
| **P-2** | **Arquitectura dirigida por eventos (Event-Driven Architecture)** | SNS/EventBridge y SQS actúan como bus central: la publicación y el consumo de eventos desacoplan entrega de exámenes, guardado de respuestas, consolidación y reporting, absorbiendo picos de carga y preservando la trazabilidad inmutable. | Disponibilidad, Rendimiento, Seguridad (no repudio) |
| **P-3** | **CQRS + Event Sourcing selectivo** | En los flujos de calificación y aprobación: los comandos generan eventos persistidos en un ledger; las consultas leen proyecciones optimizadas de bajo retardo. Se garantiza reconstrucción histórica y separación de cargas lectura/escritura. | Seguridad, Disponibilidad, Mantenibilidad |
| **P-4** | **Backend for Frontend (BFF) / Edge Gateway** | API Gateway funciona como fachada especializada para el cliente kiosk/PWA, aplicando MFA adaptativa, rate limiting y políticas por sede. Simplifica la interfaz del front-end y concentra la seguridad perimetral. | Usabilidad, Seguridad |
| **P-5** | **Serverless-first y servicios gestionados** | Cómputo mediante AWS Lambda, datos en DynamoDB/Aurora Serverless y colas gestionadas. Este estilo paga-por-uso, escala en segundos y elimina la administración de servidores, cumpliendo la restricción de no poseer datacenter propio. | Escalabilidad, Disponibilidad, Costo-eficiencia |
| **P-6** | **Patrones de resiliencia (Bulkhead, Circuit Breaker, Retry con Back-off)** | Implementados en la malla de servicios y en los SDK para aislar fallos y evitar cascadas. Los bulkheads protegen recursos críticos; los circuit breakers abren tras errores repetidos y los reintentos escalonados manejan fallos transitorios. | Disponibilidad, Rendimiento |
| **P-7** | **Hexagonal (Ports & Adapters) a nivel de microservicio** | Dentro de cada servicio se separa el núcleo de dominio de sus adaptadores (REST, eventos, persistencia), facilitando pruebas unitarias y sustitución tecnológica sin impacto externo. | Mantenibilidad, Seguridad (validaciones en el puerto) |

Estos patrones se complementan para ofrecer un sistema altamente disponible, seguro y capaz de absorber los picos estacionales de 40 000 usuarios sin comprometer la experiencia del estudiante ni la gobernanza exigida por el Ministerio. La combinación **Microservicios + EDA** resuelve los retos operativos a gran escala, mientras **CQRS/Event Sourcing** y **Hexagonal** aseguran consistencia, auditabilidad y facilidad de evolución interna. El enfoque **Serverless-first** y los **patrones de resiliencia** reducen el riesgo operativo y el costo total de propiedad, alineándose con la obligación de justificar anualmente el presupuesto.

## Estrategia Arquitectónica

La estrategia combina los siete patrones arquitectónicos adoptados (P-1 → P-7) con las tácticas definidas en el apartado 6.1 para garantizar que los cinco drivers de calidad se cumplan de forma demostrable. Se expone primero la visión macro, luego el encaje táctico por driver y, por último, el flujo de referencia que materializa esta sinergia.

* + 1. **Vista General**
* **Macro-topología**   
  *Microservicios orientados a dominio* (P-1) desplegados sobre un entorno **Serverless-first** (P-5). Cada microservicio publica y consume eventos en un **bus EDA** (P-2).
* **Canal seguro de borde**   
  Un *Backend-for-Frontend* (P-4) en API Gateway actúa como puerta única: aplica MFA, rate limiting y rutinas de validación antes de encaminar llamadas al dominio correspondiente.
* **Persistencia y auditoría**   
  Los microservicios *Grading* y *Approval* emplean **CQRS + Event Sourcing** (P-3) sobre un almacén de eventos cifrado; las proyecciones de lectura sirven al BFF con baja latencia.
* **Resiliencia transversal**   
  Patrones **Bulkhead, Circuit Breaker y Retry** (P-6) envuelven las invocaciones internas; se instrumentan mediante bibliotecas compartidas declaradas en cada Lambda.
* **Estructura interna**   
  Cada servicio adopta una **arquitectura Hexagonal** (P-7) para aislar el núcleo de dominio de sus adaptadores (REST, eventos, persistencia).
  + 1. **Táctica x Driver**

|  |  |  |
| --- | --- | --- |
| Driver | Patrones implicados | Tácticas clave |
| ****Disponibilidad**** | P-1, P-2, P-5, P-6 | Multi-AZ activo/activo, colas con DLQ, health-checks sintéticos, auto-scaling y GameDay trimestral. |
| ****Seguridad**** | P-2, P-3, P-4, P-7 | MFA adaptativa en BFF, cifrado extremo a extremo, ledger inmutable, segmentación IAM y tokenización PII. |
| ****Rendimiento / Escalabilidad**** | P-1, P-2, P-5, P-6 | Caché multinivel, procesamiento asíncrono, provisioned concurrency programada y particionamiento de datos por ExamId. |
| ****Usabilidad / Accesibilidad**** | P-4, P-5 | PWA offline, guardado incremental, carga progresiva y métricas UX en pipeline CI/CD. |
| ****Mantenibilidad**** | P-1, P-3, P-7 | Versionado de APIs, feature flags, IaC (CDK), despliegues canary y tracing distribuido con OpenTelemetry. |

* + 1. **Flujo de Referencia (Entrega y Calificación de Examen)**
* **Inicio de sesión**  
  El estudiante se autentica en el *kiosk*; el BFF valida MFA/IP y emite un **token JWT** con claims de sede y examen.
* **Descarga del examen**  
  El microservicio *Exam* recupera el formulario desde **CloudFront** (cache hit) y registra un evento ExamStarted.
* **Envió de respuestas**  
  Cada respuesta se envía en lote de 10 preguntas; el BFF publica AnswerSubmitted en **EventBridge**. El microservicio *Submission* solo “acknowledgea” y guarda en DynamoDB (write p99 < 30 ms).
* **Autocalificación / Calificación manual**  
  *Autograder* consume eventos, calcula puntajes de opción múltiple y publica AutoScoreReady. Para ensayos, *Grading* crea tareas en su cola interna; los calificadores usan su UI, que llama al BFF y genera ManualScoreReady.
* **Consolidación y proyección de lectura**  
  El microservicio *Reporting* aplica **event sourcing**; cada evento de score actualiza una proyección “ScoreReadModel” expuesta vía GraphQL al BFF.
* **Publicación del resultado**  
  El estudiante solicita “mi puntaje”; el BFF consulta la proyección. Si el score aún no está finalizado se muestra el estado *En revisión*; cuando el flujo de *Approval* culmina, aparece *Oficial*.
* **Auditoría y trazabilidad**  
  Todos los eventos se replican en un bucket S3 inmutable con ObjectLock; un job Lambda semanal verifica hashes y genera un informe de integridad para la entidad auditora.

# Vistas de Arquitectura

## Escenarios Arquitectónicos Identificados

Los siguientes escenarios representan los flujos de trabajo críticos que la arquitectura del sistema "Alcanzando la Nota" debe soportar. Estos escenarios constituyen la base para las vistas arquitectónicas y verifican que la arquitectura propuesta satisface los requisitos funcionales y no funcionales establecidos.

* + 1. **Escenario: Toma de Examen con Estrategia Multi-Modo**

**Descripción:** Un estudiante accede al sistema desde un centro educativo autorizado y realiza un examen nacional utilizando una de las tres estrategias de ejecución disponibles (dispositivo propio, o streaming remoto).

**Flujo principal:**

1. El estudiante se autentica en el sistema con credenciales y segundo factor (MFA)
2. El sistema verifica que el acceso proviene de un centro autorizado (geofencing/red interna)
3. El sistema comprueba la integridad del dispositivo mediante attestation
4. El sistema carga el examen según la estrategia configurada para ese centro
5. El estudiante visualiza y responde las preguntas con auto-guardado periódico
6. Al finalizar, el estudiante envía explícitamente el examen completo

**Atributos de calidad evaluados:** Seguridad/Rendimiento, Disponibilidad

* + 1. **Escenario: Envío y Consolidación de Respuestas**

**Descripción:** El sistema procesa y almacena las respuestas de un examen finalizado, asegurando su persistencia y disponibilidad para calificación.

**Flujo principal:**

1. El sistema recibe el envío del examen completado
2. Las respuestas de opción múltiple se almacenan directamente en DynamoDB
3. Los ensayos extensos se guardan en S3 con referencias en DynamoDB
4. El sistema emite un evento de ExamenCompletado al bus de eventos
5. El servicio de consolidación procesa el evento y actualiza las bases analíticas
6. El sistema confirma al estudiante la recepción exitosa del examen

**Atributos de calidad evaluados:** Rendimiento/Escalabilidad, Disponibilidad

* + 1. **Escenario: Calificación Manual por Calificadores**

**Descripción:** Un calificador evalúa respuestas de tipo ensayo y asigna puntuaciones con comentarios justificativos.

**Flujo principal:**

1. El calificador se autentica en el sistema con sus credenciales y MFA
2. El sistema presenta una lista de ensayos pendientes de calificación
3. El calificador accede a cada ensayo desde S3 a través de la interfaz
4. Para cada respuesta, el calificador asigna una puntuación y comentarios
5. El sistema registra la calificación y genera un evento de NotaAsignada
6. El sistema actualiza el estado del examen y notifica al servicio de consolidación

**Atributos de calidad evaluados:** Seguridad, Mantenibilidad, Rendimiento

* + 1. **Escenario: Flujo de Aprobación de Cambios de Nota**

**Descripción:** Proceso de modificación de una calificación que requiere la aprobación secuencial de tres entidades gubernamentales diferentes.

**Flujo principal:**

1. Un calificador solicita un cambio en una nota ya registrada
2. El sistema inicia un flujo de trabajo en Step Functions con la solicitud
3. La primera entidad revisa, aprueba o rechaza con justificación
4. En caso de aprobación, se notifica a la segunda entidad
5. La segunda entidad revisa y decide, notificando a la tercera
6. La tercera entidad da la aprobación final y autoriza el cambio
7. El sistema actualiza la nota y genera un registro auditable completo

**Atributos de calidad evaluados:** Seguridad, Mantenibilidad

* + 1. **Escenario: Generación de Reportes Nacionales**

**Descripción:** Un administrador del Ministerio genera reportes consolidados de resultados con múltiples criterios de filtrado para análisis educativo.

**Flujo principal:**

1. El administrador accede al módulo de reportes con autenticación reforzada
2. Define parámetros de reporte (periodo, regiones, materias, filtros específicos)
3. El sistema consulta las bases de datos analíticas optimizadas (Aurora)
4. Se procesan y agregan datos masivos con optimizaciones de rendimiento
5. El sistema genera visualizaciones y exportables (CSV/PDF)
6. Los resultados se entregan al administrador y se registra la generación

**Atributos de calidad evaluados:** Rendimiento, Escalabilidad, Seguridad

* + 1. **Escenario: Administración de Usuarios y Sedes**

**Descripción:** Un administrador gestiona la configuración de centros educativos y usuarios del sistema con sus permisos correspondientes.

**Flujo principal:**

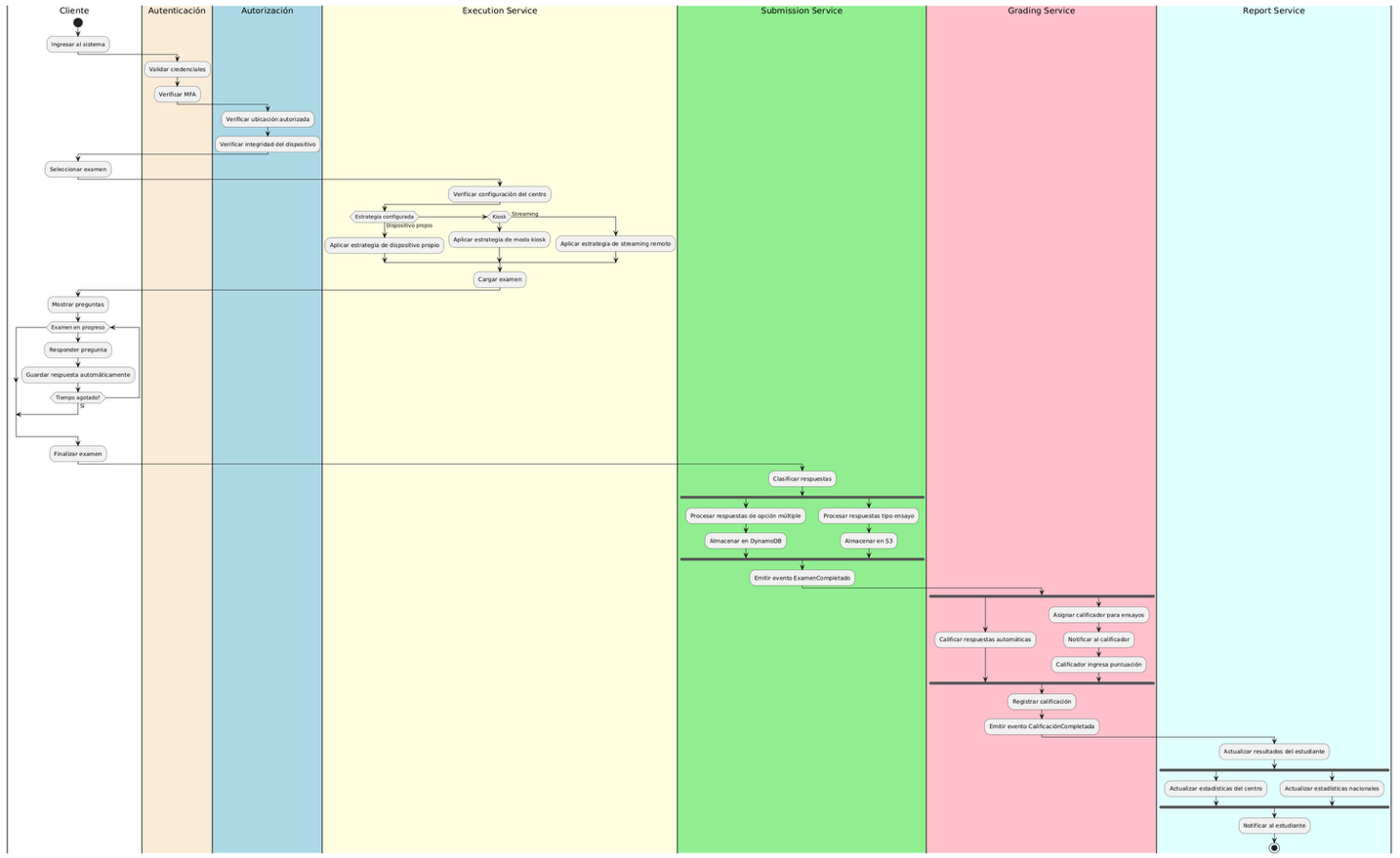
1. El administrador accede al módulo de administración
2. Crea/modifica/desactiva usuarios con sus roles específicos
3. Configura centros educativos autorizados con sus parámetros
4. Define la estrategia de ejecución de exámenes para cada centro
5. Establece ventanas de tiempo para evaluaciones nacionales
6. El sistema aplica y propaga los cambios a los servicios correspondientes

**Atributos de calidad evaluados:** Seguridad, Mantenibilidad, Disponibilidad

## Vista Lógica

* + 1. **Diagrama de Actividades del Sistema**

El siguiente diagrama de actividades representa el flujo principal del sistema "Alcanzando la Nota", mostrando el proceso de toma de examen, que es la funcionalidad central de la aplicación:



* + 1. **Descripción de la Vista**

Este diagrama de actividades representa el flujo principal del sistema "Alcanzando la Nota", mostrando la secuencia de actividades desde que un estudiante ingresa al sistema hasta que sus respuestas son procesadas, calificadas y consolidadas.

El proceso está organizado en carriles que representan los diferentes actores y servicios involucrados:

1. **Cliente**: Representa la interacción del estudiante con el sistema a través de la interfaz de usuario.
2. **Autenticación**: Maneja la validación de identidad mediante credenciales y verificación MFA.
3. **Autorización**: Verifica que el examen se realice desde un centro autorizado y con un dispositivo seguro.
4. **Execution Service**: Determina y aplica la estrategia de ejecución del examen según la configuración del centro.
5. **Submission Service**: Procesa y almacena las respuestas en los repositorios apropiados.
6. **Grading Service**: Gestiona la calificación automática y manual de las respuestas.
7. **Report Service**: Consolida los resultados y actualiza las estadísticas a todos los niveles.

El flujo incluye decisiones clave como la selección de la estrategia de ejecución y procesos paralelos como el procesamiento diferenciado de tipos de respuestas y la actualización concurrente de estadísticas.

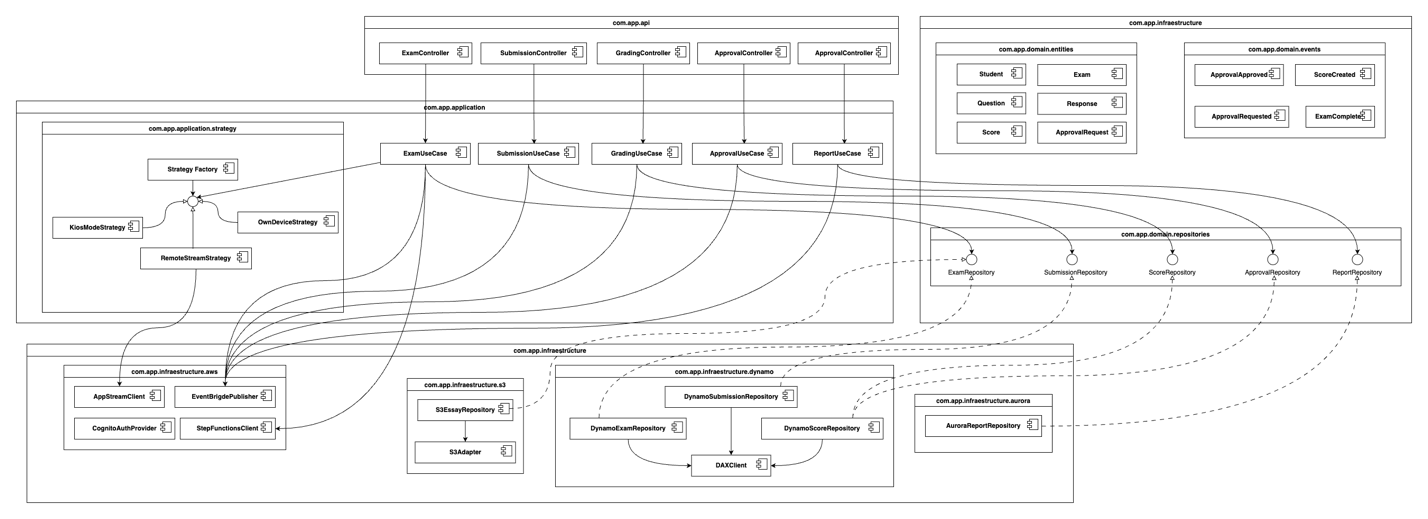
* + 1. **Catálogo de Elementos**

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Descripción** | **Rol en el Flujo** |
| **Cliente** | Interfaz de usuario para estudiantes | Permite la interacción del estudiante con el sistema de exámenes |
| **Autenticación** | Proceso de verificación de identidad | Valida que el usuario sea quien dice ser mediante credenciales y MFA |
| **Autorización** | Control de acceso basado en ubicación y dispositivo | Garantiza que el examen se realice solo en centros autorizados y con dispositivos seguros |
| **Execution Service** | Controlador de la estrategia de ejecución | Implementa el patrón Strategy para adaptar la experiencia según el modo configurado |
| **Submission Service** | Procesador de respuestas | Clasifica y almacena las respuestas en el repositorio adecuado según su tipo |
| **Grading Service** | Gestor de calificaciones | Coordina la evaluación automática y manual de las respuestas |
| **Report Service** | Consolidador de resultados | Agrega la información y actualiza estadísticas en todos los niveles |

## Vista de Implementación

* + 1. **Diagrama de Componentes**

El siguiente diagrama muestra la organización modular del código del sistema "Alcanzando la Nota", estructurado en capas y paquetes siguiendo los principios de la Arquitectura Hexagonal.



* + 1. **Descripción de la Vista**

Este diagrama de componentes muestra la organización por paquetes del sistema "Alcanzando la Nota", siguiendo una estructura vertical que refleja los principios de la Arquitectura Hexagonal (Ports & Adapters) dentro de cada dominio funcional.

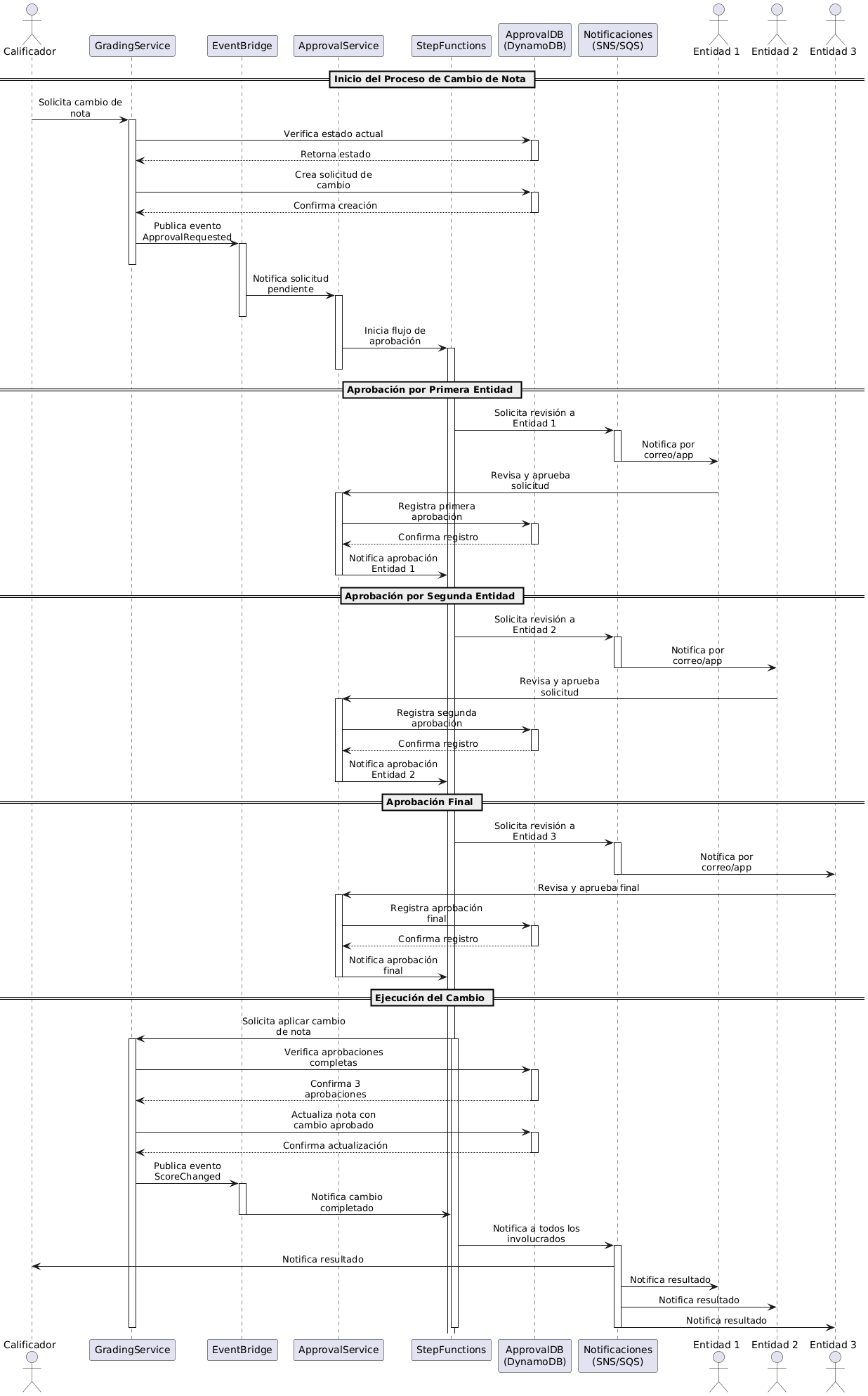
La estructura del código está organizada en cuatro capas principales, fluyendo de arriba hacia abajo:

1. **API (com.app.api)**: Contiene los controladores REST que exponen las funcionalidades del sistema a través de API Gateway. Actúan como punto de entrada para las operaciones del sistema.
2. **Aplicación (com.app.application)**: Implementa los casos de uso del sistema, orquestando las operaciones de dominio. Incluye el paquete de estrategia que implementa el patrón Strategy para las diferentes modalidades de ejecución de exámenes.
3. **Dominio (com.app.domain)**: Contiene las entidades de negocio, interfaces de repositorios (puertos secundarios) y definiciones de eventos. Esta capa es independiente de tecnologías específicas y encapsula la lógica educativa.
4. **Infraestructura (com.app.infrastructure)**: Implementa los adaptadores para interactuar con servicios AWS como DynamoDB, S3, Aurora, EventBridge y Step Functions. Cada adaptador implementa una interfaz definida en la capa de dominio.
   * 1. **Catálogo de Elementos**

|  |  |  |
| --- | --- | --- |
| **Paquete/Componente** | **Descripción** | **Propósito en la Implementación** |
| **com.app.api** | Controladores REST | Expone las funcionalidades a través de API Gateway, implementando los puertos primarios |
| **com.app.application** | Casos de uso | Orquesta el flujo de operaciones y la lógica de aplicación |
| **com.app.application.strategy** | Patrón Strategy | Permite seleccionar dinámicamente la modalidad de ejecución de exámenes |
| **com.app.domain.entities** | Objetos de dominio | Contiene las entidades del sistema con su lógica de negocio |
| **com.app.domain.repositories** | Interfaces de persistencia | Define los contratos para acceso a datos, independientes de tecnologías |
| **com.app.domain.events** | Eventos de dominio | Especifica los eventos para comunicación asíncrona entre servicios |
| **com.app.infrastructure.dynamo** | Adaptadores DynamoDB | Implementa acceso a DynamoDB para almacenamiento de alta concurrencia |
| **com.app.infrastructure.s3** | Adaptadores S3 | Gestiona el almacenamiento de ensayos extensos |
| **com.app.infrastructure.aurora** | Adaptadores Aurora | Implementa acceso a base de datos relacional para reportes |
| **com.app.infrastructure.aws** | Adaptadores AWS | Proporciona integración con EventBridge, Step Functions, Cognito y AppStream |

## Vista de Procesos

* + 1. **Diagrama de Componentes**



* + 1. **Descripción de la Vista**

Este diagrama de secuencia muestra el flujo de procesos para uno de los escenarios más críticos del sistema: el proceso de aprobación de cambios de nota que requiere la validación de tres entidades gubernamentales diferentes. El diagrama ilustra cómo se coordinan los diferentes servicios y actores a lo largo del tiempo, destacando las interacciones síncronas y asíncronas.

El proceso comienza cuando un calificador solicita un cambio en una nota ya registrada. El GradingService verifica el estado actual y crea una solicitud de cambio en la base de datos. Luego, publica un evento en EventBridge, que notifica al ApprovalService, el cual inicia un flujo de aprobación en AWS Step Functions.

El flujo de trabajo se estructura en tres fases secuenciales de aprobación, correspondientes a las tres entidades gubernamentales requeridas. Cada fase sigue un patrón similar:

1. Step Functions envía una notificación a la entidad correspondiente
2. La entidad revisa y aprueba (o rechaza) la solicitud
3. ApprovalService registra la decisión y notifica a Step Functions

Una vez que las tres entidades han aprobado la solicitud, Step Functions coordina la ejecución del cambio a través del GradingService, que verifica las aprobaciones, actualiza la nota y publica un evento de cambio completado. Finalmente, se notifica a todos los actores involucrados sobre el resultado del proceso.

Este diagrama ilustra varios aspectos importantes de la arquitectura de procesos:

* **Orquestación asíncrona**: Step Functions coordina todo el flujo sin bloquear recursos
* **Procesamiento basado en eventos**: Los servicios se comunican a través de eventos en EventBridge
* **Persistencia de estado**: DynamoDB mantiene el estado de cada solicitud de cambio
* **Notificaciones desacopladas**: El sistema de notificaciones (SNS/SQS) desacopla la comunicación con actores externos
  + 1. **Catálogo de Elementos**

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Descripción** | **Rol en el Proceso** |
| **Calificador** | Usuario del sistema que inicia la solicitud de cambio | Inicia el proceso y recibe la notificación del resultado final |
| **GradingService** | Microservicio responsable de la gestión de calificaciones | Crea la solicitud inicial, verifica aprobaciones y ejecuta el cambio de nota |
| **EventBridge** | Bus de eventos AWS | Facilita la comunicación asíncrona entre servicios mediante eventos |
| **ApprovalService** | Microservicio que gestiona el proceso de aprobación | Procesa las aprobaciones y actualiza el estado de la solicitud |
| **StepFunctions** | Servicio de orquestación de AWS | Coordina el flujo de aprobación completo como una máquina de estados |
| **ApprovalDB (DynamoDB)** | Base de datos NoSQL | Almacena el estado de las solicitudes y aprobaciones con baja latencia |
| **Notificaciones (SNS/SQS)** | Sistema de mensajería | Entrega notificaciones a las entidades gubernamentales y otros actores |
| **Entidades 1, 2 y 3** | Representantes de entidades gubernamentales | Revisan y aprueban los cambios de nota según su nivel de autoridad |

## Vista Física

* + 1. **Diagrama de Despliegue (AWS)**

El siguiente diagrama presenta la implementación física del sistema "Alcanzando la Nota" en la infraestructura de AWS, mostrando cómo los componentes lógicos y procesos descritos en las vistas anteriores se despliegan en servicios cloud específicos. La arquitectura serverless se distribuye en múltiples zonas de disponibilidad para garantizar la alta disponibilidad requerida durante los periodos de evaluación nacional.

A screenshot of a computer

AI-generated content may be incorrect.

* + 1. **Descripción de la Vista**

Este diagrama presenta la arquitectura física del sistema "Alcanzando la Nota", mostrando cómo se implementan los componentes en la infraestructura de AWS. La arquitectura está diseñada para garantizar alta disponibilidad, escalabilidad y seguridad, con despliegue en múltiples zonas de disponibilidad (AZ) dentro de una VPC dedicada.

La arquitectura sigue un enfoque serverless con los siguientes componentes principales:

* **Capa de Distribución**: CloudFront, Route 53 y WAF proporcionan distribución global, resolución DNS y protección contra ataques.
* **Capa de Entrada**: Application Load Balancer y API Gateway actúan como puntos de entrada, gestionando el tráfico hacia los microservicios.
* **Capa de Procesamiento**: Funciones Lambda implementan los microservicios por dominio (Exam, Submission, Grading, Approval, Report), desplegados en ambas zonas de disponibilidad.
* **Capa de Almacenamiento**: Combinación de DynamoDB con DAX para respuestas de exámenes, S3 para ensayos extensos, y Aurora Serverless para reportes y consolidación.
* **Capa de Coordinación**: EventBridge para comunicación basada en eventos, Step Functions para flujos de trabajo de aprobación, y SQS/SNS para colas y notificaciones.
* **Servicios Globales**: IAM para seguridad, Cognito para autenticación MFA, KMS para cifrado, y CloudWatch/X-Ray para monitoreo.
* **Servicios de Streaming**: AppStream 2.0 para implementar la estrategia de ejecución remota de exámenes.
* **DevOps & IaC**: CloudFormation/CDK para definir la infraestructura como código, y servicios de CodePipeline para implementar CI/CD con despliegues canary.

Toda la arquitectura implementa un diseño multi-AZ para alta disponibilidad, con replicación de datos entre zonas y capacidad de fail-over automático.

* + 1. **Catálogo de Elementos**

| **Componente** | **Descripción** | **Propósito en la Arquitectura** |
| --- | --- | --- |
| **VPC** | Red virtual privada | Proporciona aislamiento de red seguro para todos los componentes del sistema |
| **Zonas de Disponibilidad** | Centros de datos físicamente separados | Garantizan alta disponibilidad (99.9%) incluso ante fallos de infraestructura |
| **Subredes Públicas** | Componentes con acceso a Internet | Albergan balanceadores de carga y componentes de borde |
| **Subredes Privadas** | Componentes sin acceso directo a Internet | Contienen Lambdas y bases de datos, mejorando la seguridad |
| **CloudFront** | CDN global | Distribuye contenido estático y APIs con baja latencia globalmente |
| **Route 53** | Servicio DNS | Gestiona resolución de nombres y políticas de enrutamiento |
| **WAF** | Firewall para aplicaciones web | Protege contra ataques comunes como inyección SQL y XSS |
| **API Gateway** | Puerta de enlace para APIs | Expone los microservicios como APIs REST, implementando seguridad y throttling |
| **Lambda** | Funciones serverless | Implementan los microservicios por dominio con auto-escalado |
| **DynamoDB** | Base de datos NoSQL | Almacena respuestas de exámenes y configuraciones con alta concurrencia |
| **DAX** | Caché para DynamoDB | Acelera lecturas para mantener latencia <300ms bajo carga |
| **S3** | Almacenamiento de objetos | Guarda ensayos extensos con replicación entre regiones |
| **Aurora Serverless** | Base de datos relacional | Almacena datos consolidados para reportes, escalando automáticamente |
| **ElastiCache** | Servicio de caché en memoria | Mejora rendimiento para consultas frecuentes y metadatos |
| **EventBridge** | Bus de eventos | Facilita comunicación asíncrona entre microservicios |
| **Step Functions** | Orquestador de flujos de trabajo | Implementa el proceso de aprobación en tres entidades |
| **SQS/SNS** | Colas y notificaciones | Gestionan notificaciones y procesamiento asíncrono |
| **Cognito** | Servicio de autenticación | Implementa registro y autenticación MFA para todos los usuarios |
| **KMS** | Gestión de claves | Proporciona cifrado en tránsito y reposo para datos sensibles |
| **CloudWatch/X-Ray** | Monitoreo y trazabilidad | Supervisan el rendimiento y ayudan en la detección de problemas |
| **AppStream 2.0** | Streaming de aplicaciones | Implementa la estrategia de ejecución remota de exámenes |
| **CloudFormation/CDK** | Infraestructura como código | Define toda la infraestructura de manera declarativa |
| **CodePipeline/Build/Deploy** | Herramientas de CI/CD | Automatizan pruebas y despliegues canary |

# Directorio

## Glosario

| **Término** | **Explicación** |
| --- | --- |
| API Gateway | Servicio de AWS que actúa como "puerta de entrada" para las APIs, proporcionando un punto único de entrada para las aplicaciones cliente y gestionando aspectos como autenticación, limitación de tráfico y transformación de protocolos. |
| AppStream 2.0 | Servicio de streaming de aplicaciones de AWS que permite ejecutar aplicaciones de escritorio en un navegador web sin instalar software en el dispositivo del usuario. |
| Arquitectura Hexagonal | Patrón arquitectónico también conocido como "Ports & Adapters" que separa el núcleo de la aplicación (lógica de negocio) de los detalles técnicos externos mediante interfaces bien definidas (puertos) y sus implementaciones (adaptadores). |
| Attestation | Proceso de verificación de la integridad y autenticidad de un dispositivo o plataforma, confirmando que cumple con los requisitos de seguridad establecidos. |
| Auto-scaling | Capacidad de un sistema para aumentar o disminuir automáticamente la cantidad de recursos computacionales en función de la demanda. |
| Circuit Breaker | Patrón de diseño que previene fallos en cascada al detectar problemas en servicios dependientes y evitar llamadas innecesarias a componentes con mal funcionamiento. |
| CQRS | Patrón arquitectónico que separa las operaciones de lectura (queries) de las operaciones de escritura (commands), permitiendo optimizarlas independientemente. |
| Despliegue Canary | Estrategia de implementación que consiste en liberar una nueva versión de software a un pequeño porcentaje de usuarios antes de extenderla a toda la base de usuarios. |
| Event-driven | Enfoque arquitectónico donde los componentes se comunican mediante eventos, lo que permite un alto nivel de desacoplamiento entre servicios. |
| Geofencing | Tecnología que define límites virtuales geográficos y permite detectar cuando un dispositivo entra o sale de esas áreas predefinidas. |
| Health Check | Mecanismo para verificar periódicamente el estado de un servicio o componente del sistema, detectando problemas antes de que afecten a los usuarios. |
| Hot Partition | Situación en bases de datos NoSQL donde ciertos fragmentos de datos reciben una cantidad desproporcionadamente alta de tráfico, creando cuellos de botella. |
| Infraestructura como Código (IaC) | Práctica de gestionar y provisionar infraestructura mediante archivos de configuración versionables en lugar de procesos manuales. |
| Lazy Loading | Técnica de optimización que consiste en cargar recursos solo cuando son necesarios, en lugar de cargarlos todos al inicio. |
| Microservicios | Estilo arquitectónico que estructura una aplicación como un conjunto de servicios pequeños, autónomos y con un propósito específico. |
| Multi-AZ | Despliegue de recursos en múltiples Zonas de Disponibilidad de AWS para garantizar alta disponibilidad y tolerancia a fallos. |
| Rate Limiting | Técnica para controlar la frecuencia con la que un usuario o sistema puede acceder a un recurso, previniendo sobrecargas y ataques. |
| Serverless | Modelo de computación en la nube donde el proveedor gestiona la infraestructura, permitiendo a los desarrolladores centrarse en el código sin preocuparse por administrar servidores. |
| Sharding | Técnica de particionamiento horizontal de bases de datos que distribuye conjuntos de datos en múltiples instancias para mejorar el rendimiento y la escalabilidad. |
| Sticky Session | Configuración en balanceadores de carga que garantiza que las peticiones de un mismo usuario sean dirigidas consistentemente al mismo servidor. |
| Tokenización | Proceso de sustituir datos sensibles por identificadores no sensibles (tokens) para reducir el riesgo de exposición de información confidencial. |
| TPM | Chip especializado en hardware que almacena claves de cifrado y realiza operaciones criptográficas, proporcionando una raíz de confianza para la verificación de integridad. |
| Workflow Orchestration | Coordinación automatizada de procesos complejos que involucran múltiples pasos, decisiones y servicios, generalmente implementada como una máquina de estados. |

## Acrónimos

| **Acrónimo** | **Significado** |
| --- | --- |
| ACL | Access Control List - Lista de Control de Acceso |
| ALB | Application Load Balancer - Balanceador de Carga de Aplicaciones |
| API | Application Programming Interface - Interfaz de Programación de Aplicaciones |
| AWS | Amazon Web Services - Servicios Web de Amazon |
| AZ | Availability Zone - Zona de Disponibilidad |
| CDK | Cloud Development Kit - Kit de Desarrollo en la Nube |
| CDN | Content Delivery Network - Red de Distribución de Contenido |
| CI/CD | Continuous Integration/Continuous Deployment - Integración Continua/Despliegue Continuo |
| CQRS | Command Query Responsibility Segregation - Segregación de Responsabilidad entre Comandos y Consultas |
| DAX | DynamoDB Accelerator - Acelerador de DynamoDB |
| GSI | Global Secondary Index - Índice Secundario Global |
| IaC | Infrastructure as Code - Infraestructura como Código |
| IAM | Identity and Access Management - Gestión de Identidades y Accesos |
| KMS | Key Management Service - Servicio de Gestión de Claves |
| MFA | Multi-Factor Authentication - Autenticación Multifactor |
| RBAC | Role-Based Access Control - Control de Acceso Basado en Roles |
| RFC | Requerimiento Funcional Clave |
| RN | Restricción de Negocio |
| RT | Restricción Técnica |
| S3 | Simple Storage Service - Servicio de Almacenamiento Simple |
| SLA | Service Level Agreement - Acuerdo de Nivel de Servicio |
| SNS | Simple Notification Service - Servicio de Notificación Simple |
| SQS | Simple Queue Service - Servicio de Cola Simple |
| SRP | Single Responsibility Principle - Principio de Responsabilidad Única |
| TPM | Trusted Platform Module - Módulo de Plataforma Confiable |
| UX/UI | User Experience/User Interface - Experiencia de Usuario/Interfaz de Usuario |
| VPC | Virtual Private Cloud - Nube Privada Virtual |
| WAF | Web Application Firewall - Firewall de Aplicaciones Web |

# Apéndice

## Lista de Pruebas de Concepto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Titulo de la prueba** | **ID** | **Problema de Diseño #** | **Descripción** | **Resumen de Hallazgos** |
|  |  |  |  |  |