UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

"DESARROLLO DE UNA APLICACIÓN WEB EDUCATIVA GAMIFICADA, BASADA EN ARQUITECTURA SERVERLESS EN LA NUBE, PARA LA PREPARACIÓN PREUNIVERSITARIA DE ASPIRANTES A LA UNIVERSIDAD."

Trabajo de titulación previo a la obtención del título de Ingeniero en Tecnologías de la Información

Línea de investigación: Desarrollo, aplicación de software y cyber security (seguridad cibernética).

AUTOR:

Rafael Guillermo Lara Alvear

DIRECTOR:

MSc. Fausto Alberto Salazar Fierro, PhD

Ibarra – Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE
DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	Lara Alvear Rafael Guillermo

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UNA APLICACIÓN WEB EDUCATIVA GAMIFICADA, BASADA EN ARQUITECTURA SERVERLESS EN LA NUBE, PARA LA PREPARACIÓN PREUNIVERSITARIA DE ASPIRANTES A LA UNIVERSIDAD.
AUTOR (ES):	Lara Alvear Rafael Guillermo
FECHA: DD/MM/AAAA	31/08/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☒ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniero en Tecnologías de la Información
DIRECTOR /ASESOR:	MSc. Fausto Alberto Salazar Fierro, PhD/ MSc. Landeta López Pablo Andrés

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 31 días del mes de agosto de 2026

EL AUTOR:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
RAFAEL GUILLERMO LARA ALVEAR

……
Rafael Guillermo Lara Alvear

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 31 de agosto de 2026.

MSc. Fausto Alberto Salazar Fierro, PhD
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:
Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FAUSTO ALBERTO
SALAZAR FIERRO

MSc. Fausto Alberto Salazar Fierro, PhD

C.C: 1002172631

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “ Desarrollo de una aplicación web educativa gamificada, basada en arquitectura serverless en la nube, para la preparación preuniversitaria de aspirantes a la universidad. ” elaborado por Rafael Guillermo Lara Alvear , previo a la obtención del título del Ingeniero en Tecnologías de la Información , aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
FAUSTO ALBERTO
SALAZAR FIERRO

MSc. Fausto Alberto Salazar Fierro, PhD

C.C: 1002172631

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
PABLO ANDRES
LANDETA LOPEZ

MSc. Pablo Andrés Landeta López

C.C: 1002161055

DEDICATORIA

Dedico este trabajo a mi madre, Virginia Elena Alvear Cruz, por ser el verdadero apoyo de mi vida, el sostén frente a los problemas a lo largo de mi camino y el ejemplo más grande de amor incondicional, esfuerzo, constancia y superación. Por haber estado a mi lado durante todo este proceso, por tu lucha incansable y por haber dado todo de ti para que pudiera llegar hasta este momento. Por enseñarme que la constancia significa seguir adelante incluso cuando las fuerzas parecen acabarse, y que rendirse nunca debe ser la primera opción.

Cuando quise dejarlo todo o pensé en renunciar, estuviste ahí para recordarme que podía descansar, dormir un poco y recuperar fuerzas, pero nunca abandonar. Tu firmeza, tu exigencia y tu manera de enfrentar la vida me enseñaron a continuar aun cuando el camino se hacía difícil.

La fuerza que me caracteriza, esa necesidad de seguir adelante y de no dejarme vencer ante las dificultades, viene en gran parte de ti. De todo lo que me enseñaste con tus palabras, pero sobre todo con tu ejemplo. Muchas de las cosas que hoy soy capaz de enfrentar nacen de haber aprendido de ti a luchar, a insistir y a continuar cuando rendirse parece más fácil.

Gracias a tu amor, a tu esfuerzo y a tu apoyo pude llegar hasta aquí y alcanzar esta meta. Gran parte de lo que soy y de lo que he logrado se debe a ti, a todo lo que has hecho por mí, a cada sacrificio, a cada consejo y a cada vez que has estado a mi lado impulsándome a seguir adelante.

Este logro también lleva una parte de ti, porque soy el resultado de todo lo que durante tantos años has amado, cuidado, enseñado y ayudado a construir.

Mamá, esta meta que hoy alcanzo también es tuya.

A mi hermana, Cristina Lara Alvear, por haber sido mi compañera de lucha y mi verdadero

apoyo, por haber estado a mi lado y haber luchado conmigo codo a codo para salir adelante. Por compartir conmigo las dificultades, por sostenernos mutuamente y por no dejarme solo cuando más necesitaba de alguien a mi lado.

A mi tío, Eugenio Alvear Cruz, por ser ese amigo eterno que siempre escucha y no juzga, por sus consejos, su cariño y por estar siempre dispuesto a acompañarme.

A mi padre, Víctor Hugo Lara Suárez, por enseñarme que, a pesar de los golpes de la vida, las cosas deben mantenerse con calma y que no existen males eternos.

Este trabajo también está dedicado a mi familia ibarreña (Eulalia Alvear Cruz, Mario Granja Aguinaga, María José Granja Alvear y Mario Granja Alvear), quienes fueron mi apoyo y acogida en una ciudad casi desconocida para mí en un principio, pero que con el tiempo llegué a considerar mi segundo lugar de nacimiento.

A mis mascotas: Oso, Leo, Iris, Luna, Boby, Moly, Chiquita, Lobita, Pequeña, Maya, Polar, Mía y Canela, por haber sido compañía, calma y alegría durante este proceso y en muchos momentos difíciles de mi vida.

Dedico también este logro a mí mismo, porque muchas veces fui el mayor soporte que necesité. Por todo el dolor, el cansancio interminable y las noches sin dormir que acompañaron este proceso, pero que nunca lograron hacerme rendir. Siempre avancé.

Y al niño que alguna vez pensó que jamás lograría obtener dos ingenierías en su vida.

Rafael Guillermo Lara Alvear

AGRADECIMIENTO

Agradezco a la Universidad Técnica del Norte por brindarme la oportunidad de estudiar la carrera que quizá no habría podido seguir de no ser por la apertura de sus aulas y la formación recibida durante este camino académico.

A mis amigos Jorge Guamán y Wilmer Guairacaja, por ser esos amigos inesperados que terminaron convirtiéndose casi en hermanos y en un apoyo constante hasta el final de este proceso.

A mi amigo Sebastián Espinosa, amigo de muchos años, quien entre los mejores y peores consejos también forma parte de este logro y de muchas etapas importantes de mi vida.

A la MSc. Amada Leonor Félix Bolaños, por su enseñanza y apoyo académico.

A mi tutor, PhD. Fausto Salazar Fierro, por la guía, paciencia y acompañamiento brindados durante el desarrollo de este trabajo.

Al MSc. Víctor Caranquí, Coordinador de Ingeniería en Tecnologías de la Información, por haberme brindado apoyo y comprensión en momentos complejos de mi etapa universitaria, demostrando no solo profesionalismo, sino también calidad humana.

Y finalmente, al MSc. Pablo Landeta, a quien considero uno de los mejores profesores que he tenido a lo largo de mi carrera; no por ser el profesor más fácil, sino por ser un docente verdaderamente justo, exigente y comprometido con la enseñanza.

Rafael Guillermo Lara Alvear

ÍNDICE DE CONTENIDOS

DEDICATORIA I
AGRADECIMIENTO III
ÍNDICE DE FIGURAS XII
ÍNDICE DE TABLAS XIV
RESUMEN XVII
ABSTRACT XVIII

CAPÍTULO I

1.1 Planteamiento del Problema 1
1.2 Objetivos 4
1.2.1 Objetivo General 4
1.2.2 Objetivos Específicos 4
1.3 Alcance y delimitación 5
1.4 Justificación 5

CAPÍTULO II

2.1 Antecedentes 8
2.2 Educación Preuniversitaria 14
2.2.1 Conceptualización de la Educación Preuniversitaria 14
2.2.2 Beneficios de la Educación Preuniversitaria y sus Componentes 18
2.3 Tecnologías Educativas 21
2.3.1 Conceptualización de la Tecnología 21
2.3.2 Conceptualización de la Tecnología Educativa 23
2.4 Gamificación en el Entorno Educativo 25

2.4.1 Definición de Gamificación 25
2.4.2 Elementos de Gamificación 26
2.4.3 Beneficios de la Gamificación en el Aprendizaje 28
2.5 Aplicaciones Web Educativas 30
2.5.1 Definición de Aplicación Web 30
2.5.2 Aplicaciones Web Educativas: Concepto General 32
2.5.3 Tipos de Aplicaciones Web Educativas 33
2.6 Tecnologías para el Desarrollo de Aplicaciones Web 35
2.6.1 Arquitectura de Aplicaciones Web 36
2.6.2 Herramientas y Programas Utilizados 38
2.6.3 Entornos de Desarrollo y Bibliotecas 39
2.6.4 Bases de Datos en Aplicaciones Web 41
2.6.5 Frameworks para el Desarrollo Web 42
2.6.6 Conceptos Adicionales Clave 44
2.6.7 Tecnologías de la Nube (Cloud Computing) 45
2.7 Computación Serverless 47
2.7.1 Definición de Computación Serverless 47
2.7.2 Características de la Computación Serverless 47
2.8 Tecnologías Serverless Disponibles 48
2.8.1 Amazon Web Services (AWS) 49
2.8.2 Microsoft Azure 50
2.8.3 Gestión de Identidades y Seguridad en la Nube (OAuth 2.0 y PKCE) 51
2.9 Comparativa de Plataformas Serverless 52

2.9.1 Rendimiento de servicios IoT (broker MQTT) 52
2.9.2 Costos de servicios IoT 54
2.9.3 Rendimiento general de Máquinas Virtuales (IaaS) 55
2.9.4 Selección de Amazon Web Services (AWS) 56
2.10 Evaluación de Usabilidad 58
2.10.1 Importancia de la Usabilidad. 58
2.11 Fundamentos Metodológicos y de Diseño 59
2.11.1 Metodologías Ágiles de Desarrollo de Software. 59
2.12 El Marco de Trabajo Scrum 60
2.12.1 Roles Teóricos del Equipo. 60
2.12.2 Eventos y Artefactos de Scrum 60
2.12.3 El Product Backlog y las Historias de Usuario 61
2.12.4 Estimación de Esfuerzo: Story Points 62
2.13 Gamificación y el Marco MDA 62
2.14 Ingeniería de Software y Modelado con UML 62
2.14.1 Categorización de Diagramas UML 63
2.14.2 Modelado Funcional: Diagramas de Casos de Uso. 63
2.14.3 Modelado Dinámico: Diagramas de Secuencia y Actividad. 63
2.14.4 Modelado Estructural: Diagramas de Clases y Despliegue 64
CAPÍTULO III
3.1 Patrones de Diseño Arquitectónico 66
3.2 Arquitectura General de la Solución 67
3.3 Amazon API Gateway: Recursos y Seguridad 70
3.4 Entorno de desarrollo y herramientas de soporte 71

3.5 Estructura de artefactos del frontend 72
3.6 Estructuras de datos y persistencia 73
3.7 Desempeño, escalabilidad y costos operativos 76
3.8 Riesgos técnicos y mitigaciones 76
3.9 Glosario de siglas y términos 78
3.10 Cierre del capítulo de materiales 78
3.11 Síntesis de la Metodología Integrada 79
3.11.1 Estrategia de Búsqueda y Trazabilidad Bibliográfica 79
3.11.1.1 Cadenas de Búsqueda y Bases de Datos 79
3.11.1.2 Revistas Científicas y Autores Representativos 80
3.11.1.3 Matriz de Trazabilidad Bibliográfica 81
3.12 Requerimientos del Sistema 82
3.12.1 Requerimientos Funcionales (RF). 82
3.12.2 Requerimientos No Funcionales (RNF) 83
3.13 Modelado Funcional: Especificación de Casos de Uso 84
3.13.1 Identificación de Actores 84
3.14 Casos de Uso 85

3.14.1 CU-01: Autenticación Segura y Delegada 85
3.14.2 CU-02: Registro de Auditoría de Accesos 87
3.14.3 CU-03: Selección de Área y Validación de Reglas 89
3.14.4 CU-04: Ejecución del Motor de Evaluación 92
3.14.5 CU-05: Calificación y Persistencia de Resultados 95
3.14.6 CU-06: Clasificación Global Dinámica 98
3.14.7 CU-07: Monitoreo Analítico del Administrador 100
3.14.8 CU-08: Gestión de Reactivos vía JSON 102
3.15 Modelado Dinámico: Diagramas de Secuencia 104
3.15.1 Secuencia de Evaluación y Persistencia 105
3.15.2 Secuencia de Auditoría de Seguridad 105
3.16 Modelado Estructural: Modelo Lógico NoSQL 106
3.17 Ejecución Ágil: Adaptación, Roles y Ciclo de Vida 108

3.17.1 Estructura del Equipo y Asignación de Roles 108
3.17.2 Matriz de Responsabilidades (RACI) 109
3.17.3 Fases del Proyecto y Cronograma Macro 110
3.17.4 Registro de Eventos Scrum y Tutorías Académicas 111
3.17.5 Artefactos de Scrum: Product Backlog Priorizado 113
3.17.6 Especificación de Historias de Usuario y Criterios de Aceptación 116
3.17.7 Gestión Visual del Backlog 123
3.17.8 Sprint Backlog y Ciclo de Desarrollo Técnico 123
3.17.9 Planificación de Tareas y Estimación de Esfuerzo (Sprints 1 al 9) 126
3.17.10 Métricas de Control: Velocidad y Gráfico de Burndown 132
3.18 Trazabilidad del Diseño de Gamificación (MDA) 135
3.18.1 Trazabilidad Pedagógica 136
3.19 Metodología de Validación: Escala de Usabilidad del Sistema (SUS) 137
3.19.1 Fundamentación del Instrumento 137
3.19.2 Protocolo de Aplicación con Usuarios Reales 138
3.19.3 Algoritmo de Calificación y Escalamiento 139
3.19.4 Criterios de Interpretación de Resultados 139
CAPÍTULO IV
4.1 Análisis de Fiabilidad y Normalidad 141
4.1.1 Prueba de Normalidad (Kolmogorov-Smirnov) 141
4.2 Análisis e Interpretación de Resultados de Usabilidad 143

4.2.1 Población y Muestra 143
4.2.2 Tabulación de Frecuencias de la Escala SUS 143
4.2.3 Análisis Descriptivo por Preguntas 145
4.2.4 Calificación Global de Usabilidad y Aceptabilidad 156
4.2.5 Validación de Fiabilidad del Instrumento (Alfa de Cronbach) 157
4.3 Análisis de Impactos del Proyecto 158
4.3.1 Impacto Económico. 159
4.3.2 Impacto Socio Cultural 162
4.3.3 Impacto Tecnológico y Ambiental 163
4.3.4 Impacto Ético y Legal 164
4.3.5 Nivel de Impacto Global del Proyecto 165
CONCLUSIONES 167
RECOMENDACIONES 168
BIBLIOGRAFÍA 169
ANEXOS 176
4.4 Anexo A.- Formato y Métricas del Instrumento de Evaluación (Google Forms) 176
4.5 Anexo B.- Diagrama de Interacción Orientado a Eventos (FaaS) 178
4.6 Anexo C.- Interfaces de la Aplicación *PreServerless* 180
4.6.1 C.1. Módulo de Gestión Administrativa 180
4.6.2 C.2. Módulo del Estudiante (Evaluación y Gamificación) 183
4.7 Anexo D.- Matriz de Tabulación y Validación Estadística 185

4.7.1 D.1. Evidencia de Procesamiento en Entorno de Desarrollo 185

4.7.2 D.2. Resumen Matemático de Fiabilidad 185

4.7.3 D.3. Matriz de Datos Consolidada (Exportación) 186

ÍNDICE DE FIGURAS

Figura 1 Arquitectura *serverless* implementada en AWS para la aplicación web educativa gamificada. 68
Figura 2 Diagrama de Despliegue UML de la infraestructura Serverless. 70
Figura 3 Caso de uso Autenticación 85
Figura 4 Caso de uso Registro de Auditoría 87
Figura 5 Caso de uso Selección de Área 90
Figura 6 Caso de uso Ejecución de Evaluación 92
Figura 7 Diagrama de Actividades UML: Lógica concurrente del motor de evaluación. 95
Figura 8 Caso de uso Calificación de Resultados 96
Figura 9 Caso de uso Clasificación Global 98
Figura 10 Caso de uso Monitoreo Analítico 100
Figura 11 Caso de uso Gestión de Reactivos 102
Figura 12 Diagrama de secuencia: Ejecución y calificación del simulacro. 105
Figura 13 Diagrama de secuencia: Registro de trazabilidad y accesos. 106
Figura 14 Modelo lógico de datos en Amazon DynamoDB. 107
Figura 15 Tablero Kanban utilizado para la gestión visual del Product Backlog. . 123
Figura 16 Gráfico de velocidad: Puntos planificados vs. completados por cada Sprint. 134
Figura 17 Gráfico de Burndown del Sprint 8: Seguimiento diario del trabajo pendiente. 135

Figura 18 Distribución empírica del Puntaje Global SUS frente a la curva normal teórica. 142
Figura 19 Distribución de Frecuencias por Reactivo - Instrumento SUS (N=50) . 145
Figura 20 Distribución de respuestas para la Pregunta 1. 146
Figura 21 Distribución de respuestas para la Pregunta 2. 147
Figura 22 Distribución de respuestas para la Pregunta 3. 148
Figura 23 Distribución de respuestas para la Pregunta 4. 149
Figura 24 Distribución de respuestas para la Pregunta 5. 150
Figura 25 Distribución de respuestas para la Pregunta 6. 151
Figura 26 Distribución de respuestas para la Pregunta 7. 152
Figura 27 Distribución de respuestas para la Pregunta 8. 153
Figura 28 Distribución de respuestas para la Pregunta 9. 154
Figura 29 Distribución de respuestas para la Pregunta 10. 155
Figura 30 Encabezado del formulario, objetivos de la investigación e instrucciones para el usuario. 176
Figura 31 Sección de reactivos de usabilidad empleando la escala Likert de cinco puntos. 176
Figura 32 Métricas de recolección de datos y resumen automatizado de participación generadas por la herramienta. 177
Figura 33 Diagrama de secuencia de invocaciones FaaS y lógica de negocio en la infraestructura AWS. 179
Figura 34 Acceso inicial al sistema. 180

Figura 35 Portal de autenticación y control de seguridad gestionado por AWS Cognito. 180
Figura 36 Panel de navegación principal con control de acceso basado en roles (RBAC). 181
Figura 37 Módulo de Inteligencia Educativa para el análisis estadístico de rendimientos. 181
Figura 38 Gestión de persistencia NoSQL: Carga manual y masiva de reactivos mediante JSON. 182
Figura 39 Documento de reporte analítico consolidado generado automáticamente en formato PDF. 182
Figura 40 Selección de áreas de conocimiento y diagnóstico inicial para el aspirante. 183
Figura 41 Entorno de ejecución del simulacro con carga dinámica de reactivos y control de tiempo. 183
Figura 42 Visualización del Ranking Global y mecánicas de gamificación (*Leaderboard*). 184
Figura 43 Script de automatización estadística y salida en consola (Alfa = 0.821). 185
Figura 44 Tabulación de respuestas en Excel. 187

ÍNDICE DE TABLAS

Tabla II	Antecedentes: síntesis de evidencias y relevancia	12
Tabla III	Educación preuniversitaria: conceptos y alcance	15
Tabla IV	Elementos fundamentales en la educación preuniversitaria	16
Tabla V	Beneficios de la educación preuniversitaria y sus componentes	19
Tabla VI	Síntesis de conceptualizaciones de tecnología	22
Tabla VII	Síntesis de conceptualizaciones de tecnología educativa	24
Tabla VIII	Definiciones y alcance de la gamificación	25
Tabla IX	Elementos de gamificación y propósito educativo	27
Tabla X	Beneficios de la gamificación en educación	29
Tabla XI	Aplicación web: estructura y ventajas en educación	31
Tabla XII	Aplicaciones web educativas: objetivos y aportes	32
Tabla XIII	Tipos comunes de aplicaciones web educativas	33
Tabla XIV	Tecnologías base para aplicaciones web	35
Tabla XV	Modelos y patrones arquitectónicos	37
Tabla XVI	Herramientas de plataforma y soporte	38
Tabla XVII	Entornos y bibliotecas destacadas	40
Tabla XVIII	Prácticas de persistencia y automatización	41
Tabla XIX	Frameworks y *runtimes* relevantes	42
Tabla XX	Estrategias de generación de código	43
Tabla XXI	Conceptos adicionales: síntesis operativa	44
Tabla XXII	Modelos y capacidades en *cloud computing*	46
Tabla XXIII	Características clave del modelo *serverless*	47
Tabla XXIV	Servicios AWS relevantes para una arquitectura *serverless* educativa	49
Tabla XXV	Rendimiento MQTT gestionado por proveedor en escenarios típicos	53
Tabla XXVI	Resumen comparativo de costos en IoT gestionado	54
Tabla XXVII	Desempeño relativo de VM por proveedor y dimensión	56
Tabla XXVIII	Criterios de selección de AWS y sustento	57
Tabla XXIX	Importancia de la usabilidad en soluciones *cloud*/*serverless*	58
Tabla XXX	Resumen de diagramas UML aplicados según Jiménez de Parga	64
Tabla XXXI	Componentes principales de la arquitectura *serverless*	68
Tabla XXXII	Recursos REST publicados en Amazon API Gateway	71
Tabla XXXIII	Herramientas de desarrollo, librerías y propósito	72
Tabla XXXIV	Organización de artefactos del frontend	72
Tabla XXXV	Campos lógicos del ítem de pregunta para carga en JSON	73
Tabla XXXVI	Estructura lógica de la tabla de resultados (`resultados`)	74
Tabla XXXVII	Estructura lógica de la tabla de auditoría de accesos (`logins`)	75
Tabla XXXVIII	Consideraciones de rendimiento y escalabilidad	76
Tabla XXXIX	Principales riesgos y acciones de mitigación	77
Tabla XL	Glosario de términos técnicos utilizados	78
Tabla XLI	Cadenas de búsqueda booleanas aplicadas por área de estudio	80
Tabla XLII	Trazabilidad: Del Marco Conceptual a la Arquitectura Funcional	81
Tabla XLIII	Matriz de Requerimientos Funcionales	82
Tabla XLIV	Matriz de Requerimientos No Funcionales	83
Tabla XLV	Especificación de CU-01: Autenticación Segura y Delegada	85

Tabla XLVI Flujo Alternativo 1.1: Credenciales Inválidas . . . 86
Tabla XLVII Especificación de CU-02: Registro de Auditoría de Accesos . . . 88
Tabla XLVIII Flujo Alternativo 2.1: Fallo de persistencia . . . 89
Tabla XLIX Especificación de CU-03: Selección de Área y Validación de Reglas 90
Tabla L Flujo Alternativo 3.1: Límite de intentos alcanzado . . . 91
Tabla LI Especificación de CU-04: Ejecución del Motor de Evaluación . . . 93
Tabla LII Flujo Alternativo 4.1: Tiempo agotado . . . 94
Tabla LIII Especificación de CU-05: Calificación y Persistencia de Resultados . 96
Tabla LIV Flujo Alternativo 5.1: Falla de conexión . . . 97
Tabla LV Especificación de CU-06: Clasificación Global Dinámica . . . 99
Tabla LVI Especificación de CU-07: Monitoreo Analítico del Administrador . . 100
Tabla LVII Flujo Alternativo 7.1: Exportar PDF . . . 101
Tabla LVIII Especificación de CU-08: Gestión de Reactivos vía JSON . . . 103
Tabla LIX Flujo Alternativo 8.1: JSON Inválido . . . 104
Tabla LX Roles de Scrum adaptados a la estructura del proyecto . . . 108
Tabla LXI Matriz RACI para actividades clave del ciclo de vida del software . . 109
Tabla LXII Cronograma Macro de Fases de Desarrollo y Redacción . . . 110
Tabla LXIII Registro de eventos iterativos, revisiones y tutorías académicas . . . 111
Tabla LXIV Product Backlog Priorizado con Criterios de Aceptación Serverless . 114
Tabla LXV Detalle de HU-01: Autenticación Segura y Auditoría . . . 116
Tabla LXVI Detalle de HU-02: Generación Dinámica de Reactivos (MCQ) . . . 117
Tabla LXVII Detalle de HU-03: Motor de Completación . . . 117
Tabla LXVIII Detalle de HU-04: Motor de Emparejamiento (Matching) . . . 118
Tabla LXIX Detalle de HU-05: Progresión y Bloqueo de Niveles . . . 119
Tabla LXX Detalle de HU-06: Interfaz HUD y Temporizador . . . 119
Tabla LXXI Detalle de HU-07: Clasificación Global (Leaderboard) . . . 120
Tabla LXXII Detalle de HU-08: Sistema de Medallas y Estatus . . . 121
Tabla LXXIII Detalle de HU-09: Panel Analítico del Administrador . . . 121
Tabla LXXIV Detalle de HU-10: Validación Científica de Usabilidad . . . 122
Tabla LXXV Sprint Backlog detallado por iteración (Septiembre 2025 – Marzo 2026) . . . 124
Tabla LXXVI Detalle de Tareas: Sprint 1 (Infraestructura Base) . . . 126
Tabla LXXVII Detalle de Tareas: Sprint 2 (Autenticación y Auditoría) . . . 127
Tabla LXXVIII Detalle de Tareas: Sprint 3 (Motor de Evaluación MCQ) . . . 128
Tabla LXXIX Detalle de Tareas: Sprint 4 (Lógica de Gamificación Central) . . . 128
Tabla LXXX Detalle de Tareas: Sprint 5 (Motor de Completación) . . . 129
Tabla LXXXI Detalle de Tareas: Sprint 6 (Reactivos de Emparejamiento) . . . 130
Tabla LXXXII Detalle de Tareas: Sprint 7 (Gamificación y Ranking) . . . 130
Tabla LXXXIII Detalle de Tareas: Sprint 8 (Analítica Administrativa) . . . 131
Tabla LXXXIV Detalle de Tareas: Sprint 9 (Piloto en Producción y Validación) . . . 132
Tabla LXXXV Velocidad de desarrollo por Sprint (Puntos Planificados vs. Completados) . . . 132
Tabla LXXXVI Registro de Burndown del Sprint 8 (Puntos restantes por día) . . . 133
Tabla LXXXVII Matriz de trazabilidad MDA aplicada a la plataforma . . . 135
Tabla LXXXVIII Mapeo de objetivos de aprendizaje con mecánicas de gamificación . 136
Tabla LXXXIX Ítems de la escala SUS aplicados a la plataforma PreServerless . . . 137
Tabla XC Rangos de interpretación para la puntuación SUS . . . 140

Tabla XCI Prueba de Normalidad Kolmogorov-Smirnov para Escala SUS . . . 142
Tabla XCII Distribución de frecuencias de la escala SUS ($n = 50$). . . . 143
Tabla XCIII Frecuencias y porcentajes de la Pregunta 1 . . . 145
Tabla XCIV Frecuencias y porcentajes de la Pregunta 2 . . . 147
Tabla XCV Frecuencias y porcentajes de la Pregunta 3 . . . 148

Tabla XCVI Frecuencias y porcentajes de la Pregunta 4 149
Tabla XCVII Frecuencias y porcentajes de la Pregunta 5 150
Tabla XCVIII Frecuencias y porcentajes de la Pregunta 6 151
Tabla XCIX Frecuencias y porcentajes de la Pregunta 7 152
Tabla C Frecuencias y porcentajes de la Pregunta 8 153
Tabla CI Frecuencias y porcentajes de la Pregunta 9 154
Tabla CII Frecuencias y porcentajes de la Pregunta 10 155
Tabla CIII Resultados globales de la evaluación SUS 156
Tabla CIV Interpretación lógica de puntajes SUS (Adaptado de Bangor et al., 2009) 157
Tabla CV Resultados del análisis de fiabilidad de la muestra 158
Tabla CVI Valores de referencia para el análisis de impacto 159
Tabla CVII Matriz de Impacto Económico 159
Tabla CVIII Comparativa de Eficiencia: Monolito vs. PreServerless 161
Tabla CIX Comparativa de Capa Gratuita AWS vs. Consumo del Proyecto . . . 162
Tabla CX Matriz de Impacto Socio Cultural 163
Tabla CXI Matriz de Impacto Tecnológico y Ambiental 164
Tabla CXII Matriz de Impacto Ético y Legal 165
Tabla CXIII Resumen del Nivel de Impacto Global 166

RESUMEN

El acceso limitado a herramientas digitales educativas y el uso de metodologías tradicionales continúan representando una dificultad en los procesos de preparación preuniversitaria, especialmente en entornos donde el acceso a recursos tecnológicos sigue siendo reducido. Como respuesta a esta problemática, se desarrolló una aplicación web educativa basada en arquitectura serverless en la nube, orientada a la ejecución de simulaciones y evaluaciones interactivas para aspirantes a la educación superior. La solución fue implementada mediante servicios de Amazon Web Services (AWS) bajo un modelo orientado a eventos, permitiendo disponer de una plataforma escalable, flexible y con bajo consumo de recursos de infraestructura mediante el aprovechamiento de servicios administrados y la capa gratuita del proveedor cloud. El sistema incorpora funcionalidades como autenticación segura, generación dinámica de reactivos, almacenamiento de resultados en bases de datos NoSQL, clasificación global y progresión por niveles mediante elementos de gamificación integrados en el entorno de evaluación. Además, se aplicó la metodología Scrum para la planificación y desarrollo iterativo del proyecto. La evaluación de usabilidad fue realizada mediante la escala System Usability Scale (SUS) con una muestra de 50 usuarios, obteniéndose resultados favorables relacionados con la facilidad de uso, interacción y aceptación general de la plataforma desarrollada.

Palabras clave: arquitectura serverless, aplicación web, Amazon Web Services, gamificación, usabilidad

ABSTRACT

Limited access to digital educational tools and the use of traditional methodologies continue to represent a difficulty in pre-university preparation processes, especially in environments where access to technological resources remains limited. In response to this problem, an educational web application based on a cloud serverless architecture was developed for the execution of simulations and interactive assessments aimed at students applying to higher education institutions. The solution was implemented using Amazon Web Services (AWS) under an event-driven model, enabling the deployment of a scalable and flexible platform with low infrastructure resource consumption through the use of managed services and the cloud provider free tier. The system incorporates functionalities such as secure authentication, dynamic question generation, result storage in NoSQL databases, global ranking, and level progression through integrated gamification elements within the assessment environment. In addition, the Scrum methodology was applied for project planning and iterative development. Usability evaluation was conducted through the System Usability Scale (SUS) with a sample of 50 users, obtaining favorable results regarding ease of use, interaction, and overall acceptance of the developed platform.

Keywords: serverless architecture, web application, Amazon Web Services, gamification, usability

CAPÍTULO I

INTRODUCCIÓN

1.1 Planteamiento del Problema

Cada año, cientos de estudiantes en el país enfrentan el desafío de prepararse adecuadamente para los exámenes de ingreso a la universidad. Sin embargo, una parte considerable de estos jóvenes no cuenta con herramientas digitales eficaces ni con metodologías que realmente estimulen su implicación con el aprendizaje. Las técnicas de estudio convencionales, centradas en la memorización y sin interacción significativa, suelen resultar monótonas y poco eficaces para los estudiantes actuales, generando desinterés y bajo rendimiento académico [1].

El principal problema que se aborda se basa en la gran brecha digital que aún existe en las instituciones educativas secundarias en el Ecuador. Esto se evidencia en la desigualdad en el acceso y uso de herramientas y recursos tecnológicos, identificándose una mayor desigualdad entre instituciones privadas y públicas, así como también entre zonas urbanas y rurales del país. Nos encontramos con un desafío que incluye a docentes con disminuida suficiencia en habilidades tecnológicas y en la integración de las tecnologías de la información y comunicación (TIC) de forma eficiente en el transcurso del proceso de enseñanza-aprendizaje. Esto limita de forma significativa las oportunidades para los estudiantes que no pueden participar en entornos educativos más digitales, afectando también directamente la calidad de educación que obtienen. Este problema fue más evidente durante la pandemia del COVID-19, aumentando estas disparidades ya existentes.[2].

Otra problemática abarca la amplia disparidad entre la cantidad de cupos disponibles y la alta demanda de acceso a la educación superior en Ecuador, aunque la oferta ha tenido un aumento considerable. Esto trae a su vez otro problema: la diversidad de conocimientos y los distintos

tipos de jóvenes aspirantes con diferentes estatus socioeconómicos, lo cual es una traba más tanto en el proceso de ingreso a las universidades e instituciones superiores como en el proceso de nivelación académica, lo que provoca una alta tasa de deserción estudiantil. Esto se agrava debido a las limitaciones de infraestructura, tecnología y nivel académico de las instituciones superiores, incluyendo laboratorios de TI con buena conectividad[3].

Desde el campo de la Ingeniería en Tecnologías de la Información, esta realidad plantea una oportunidad concreta para proponer una solución tecnológica que aproveche modelos innovadores como la gamificación y los beneficios de la computación sin servidor. En particular, la arquitectura serverless representa una opción moderna, eficiente y escalable para construir aplicaciones web accesibles sin la necesidad de administrar infraestructura física o virtual [4], [5].

Este enfoque permite implementar funcionalidades bajo demanda, manteniendo costos bajos y reduciendo la complejidad técnica, lo que lo convierte en un modelo adecuado para sistemas educativos de amplio alcance. Servicios como AWS Lambda, por ejemplo, permiten ejecutar código automáticamente en respuesta a eventos, lo que se adapta perfectamente a las interacciones propias de una aplicación educativa en línea [6].

Además, la integración de bases de datos NoSQL como Amazon DynamoDB facilita la gestión eficiente de contenidos y registros de usuarios, con alta disponibilidad y respuesta en tiempo real [7].

Causas	**Efectos**	**Referencias**
Falta de acceso a tecnologías educativas adaptativas en zonas rurales.	Bajo nivel de preparación para los exámenes de ingreso a la universidad.	Muñoz et al. [2]
Uso predominante de metodologías tradicionales no interactivas.	Baja participación y escasa comprensión de contenidos complejos.	Li et al. [1]

Ausencia de herramientas de autoevaluación con retroalimentación inmediata.	Imposibilidad de identificar debilidades específicas en el aprendizaje.	Lampropoulo et al. [8]
Limitaciones técnicas de arquitecturas convencionales (costos, escalabilidad).	Dificultad para sostener aplicaciones educativas accesibles y funcionales a gran escala.	Hassan et al. [4]
Falta de adaptabilidad del sistema a demandas variables de estudiantes.	Sobrecargas en recursos tecnológicos durante épocas de mayor uso (como períodos de admisión).	Jonas et al. [5]
Inexistencia de aplicaciones basadas en modelos *serverless* para el segmento preuniversitario.	Oportunidad desaprovechada para optimizar el acceso, mantenimiento y escalabilidad en entornos educativos.	AWS Lambda Docs [6]
Falta de integración de bases de datos escalables para manejar grandes volúmenes de datos de estudiantes.	Limitaciones en el seguimiento académico y almacenamiento de grandes volúmenes de datos.	AWS DynamoDB Docs [7]
Necesidad de modelos arquitectónicos más flexibles para aplicaciones educativas.	Alta complejidad para actualizar y mantener sistemas educativos grandes.	Ortiz [9]
Falta de herramientas adecuadas para medir la percepción del usuario.	Dificultad para validar la experiencia del usuario y mejorar las interfaces educativas.	Brooke [10]

Falta de enfoques tecnológicos innovadores para la educación preuniversitaria.	Oportunidades desaprovechadas para mejorar los resultados académicos.	PreuApp [11]

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar una aplicación web educativa gamificada, basada en arquitectura serverless en la nube, para la preparación preuniversitaria de aspirantes a la universidad.

1.2.2 Objetivos Específicos

- Establecer un marco conceptual actualizado sobre serverless computing, aplicaciones educativas web y elementos de gamificación funcional aplicados a cuestionarios digitales, para sustentar el diseño de la solución propuesta.

- Diseñar la arquitectura funcional de la aplicación web, integrando servicios cloud bajo el modelo serverless (FaaS, bases de datos administradas, almacenamiento escalable).

- Desarrollar una aplicación web que incluya elementos de gamificación basada en serverless computing.

- Evaluar la usabilidad de la aplicación desarrollada aplicando la encuesta System Usability Scale (SUS).

1.3 Alcance y delimitación

Este proyecto abarcará desde la concepción técnica hasta la implementación de una aplicación web funcional utilizando serverless computing, orientada al apoyo en la preparación preuniversitaria. La solución se desarrollará de forma modular e incluirá funcionalidades básicas para la gestión de contenidos educativos y evaluación interactiva mediante cuestionarios de completación, ejercicios de unión de columnas y pruebas de ubicación por nivel. Estas funcionalidades permitirán organizar los contenidos conforme a la complejidad de los ejercicios, facilitando así una experiencia personalizada. La arquitectura estará diseñada bajo el modelo serverless, utilizando servicios gestionados en la nube como funciones FaaS, bases de datos administradas y almacenamiento escalable. Esto permitirá un despliegue flexible, con capacidad de escalar automáticamente según el número de usuarios, reduciendo la necesidad de gestión de infraestructura. La aplicación estará disponible para usuarios a través de un entorno cloud público, y se contemplará su validación mediante pruebas con estudiantes reales. Para ello, se utilizará el System Usability Scale (SUS) como método de evaluación estandarizado, permitiendo obtener una valoración objetiva sobre la usabilidad del sistema desde la perspectiva del usuario. Elementos como funcionalidades multilenguaje, expansión a nuevas materias o mantenimientos a largo plazo no forman parte de este alcance inmediato, aunque podrían considerarse en futuras iteraciones del proyecto.

1.4 Justificación

Los estudiantes en los últimos años del bachillerato pasan por un momento crucial para el ingreso a las universidades. En la mayoría de los casos, esto se da sin ninguna supervisión ni apoyo de recursos tecnológicos que sean accesibles y digitales, lo que ocasiona un aumento de la desigualdad entre los estudiantes con menos limitaciones económicas y geográficas[8].

Frente a esta realidad, surge la necesidad de construir herramientas que se adapten a las condiciones reales de los usuarios y que puedan ser sostenibles técnica y económicamente. En

ese sentido, la presente propuesta busca diseñar e implementar una aplicación web funcional utilizando computación sin servidor (serverless computing), lo que implica construir un sistema donde el control de la infraestructura no recae en el desarrollador, sino en el proveedor de servicios cloud. Esta arquitectura elimina la necesidad de administrar servidores físicos o virtuales y, en su lugar, permite ejecutar funciones específicas como respuesta a eventos, con alta escalabilidad y menor complejidad de mantenimiento [4], [5].

Uno de los pilares técnicos del sistema será AWS Lambda, un servicio que permite desplegar código sin necesidad de configurar entornos completos, facilitando la implementación de funcionalidades críticas como el registro de usuarios, el procesamiento de respuestas o la retroalimentación instantánea [6]. Este enfoque será complementado por Amazon DynamoDB, una base de datos NoSQL que proporciona almacenamiento persistente con alta disponibilidad, respuesta rápida y escalado automático, adaptándose perfectamente al comportamiento de una aplicación educativa que podría registrar variaciones de uso según la temporada académica [7].

Desde el punto de vista pedagógico, el componente de gamificación será aplicado de forma puntual y medible, centrado únicamente en ejercicios de completación y asociación de columnas, así como un sistema de niveles que organice progresivamente la dificultad de los contenidos. Esta estructura permitirá al usuario avanzar en función del dominio de conceptos clave sin requerir un entorno lúdico complejo, facilitando la validación del diseño en escenarios reales.

Una de las ventajas clave de este proyecto es la elección de una arquitectura basada en servicios gestionados, lo que permite reducir significativamente los costos operativos. Al trabajar bajo un modelo de pago por uso, el sistema solo consume recursos cuando realmente se los necesita, lo que lo hace escalable, eficiente y viable tanto para entornos de prueba como para su implementación en escenarios reales[10].

En cuanto al componente formativo, este proyecto contribuye al desarrollo de competencias técnicas en el uso de tecnologías actuales. Se trabaja con soluciones cloud native, patrones orientados a eventos (event-driven) y herramientas modernas de backend como FaaS (Functions as a Service), que permiten ejecutar funciones individuales en la nube sin necesidad de gestionar

servidores. También se hace uso de bases de datos serverless, que se ajustan automáticamente a la demanda[10].

Todo esto permite al estudiante aplicar conceptos prácticos en un entorno real y actual, mientras se mantiene alineado con la línea de investigación institucional sobre tecnologías aplicadas a la educación. La propuesta resulta así replicable y útil para contextos similares, aportando una alternativa concreta frente a los desafíos tecnológicos que enfrenta el sistema educativo[2], [9].

Finalmente, se prevé la utilización del System Usability Scale (SUS) como instrumento de evaluación de la experiencia del usuario. Esta herramienta estandarizada ha sido ampliamente adoptada para medir la usabilidad de sistemas, mediante una escala de 10 ítems cuya valoración permite establecer niveles de aceptación sobre la facilidad de uso percibida por los usuarios finales[10].

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Durante los últimos años, se han desarrollado diversos estudios que confirman la utilidad de enfoques tecnológicos como la computación serverless y la gamificación en contextos educativos digitales. Estos trabajos han servido de base para delimitar la propuesta actual, centrada en el desarrollo de una aplicación web funcional utilizando serverless computing y técnicas específicas de gamificación aplicadas a la preparación preuniversitaria.

Un estudio especialmente revelador sobre el impacto de la gamificación en la educación es el realizado por Lampropoulos y Sidiropoulos. Los autores analizaron el desempeño de más de mil estudiantes universitarios expuestos a tres modelos de aprendizaje distintos: uno tradicional, otro completamente virtual y un tercero enriquecido con dinámicas de gamificación. Los resultados fueron difíciles de ignorar. Los estudiantes que participaron en entornos gamificados no solo obtuvieron mejores calificaciones, sino que además duplicaron los niveles de rendimiento académico sobresaliente frente a los otros grupos. Paradójicamente, aquello que durante años fue visto por algunos sectores como un simple "juego" terminó funcionando como un motor de compromiso y aprendizaje mucho más eficaz que varios métodos convencionales. La diferencia entre memorizar contenidos y querer avanzar en ellos resultó, en este caso, tan evidente como la distancia entre cumplir una obligación y aceptar un desafío estimulante. Todo ello sugiere que, aplicada con criterio pedagógico, la gamificación puede convertirse en una herramienta capaz de fortalecer la motivación y mejorar significativamente el proceso de aprendizaje [8].

Complementando esta perspectiva, el metaanálisis realizado por Li, Ma y Shi, basado en 41 estudios sobre gamificación educativa, identificó un efecto positivo consistente en el uso de

dinámicas gamificadas dentro de los procesos de enseñanza, alcanzando un tamaño del efecto elevado de 0.82. Los autores destacan que elementos como los retos, los sistemas de puntuación y la progresión por niveles no solo incrementan la participación del estudiante, sino que transforman la experiencia de aprendizaje en un proceso más inmersivo y estimulante. Curiosamente, mecanismos que durante mucho tiempo fueron asociados únicamente al entretenimiento terminan desempeñando un papel pedagógico de gran eficacia. La lógica del avance gradual, casi como subir peldaños visibles dentro de una escalera, ofrece al estudiante objetivos claros y una sensación constante de progreso, algo que muchos modelos tradicionales rara vez consiguen transmitir. En consecuencia, la gamificación aparece no solo como un recurso atractivo, sino como una estrategia capaz de adaptarse con mayor sensibilidad a las necesidades y motivaciones del aprendizaje contemporáneo [1].

Desde el enfoque tecnológico, Hassan, Barakat y Sarhan llevaron a cabo una revisión exhaustiva sobre el modelo serverless computing. En sus hallazgos se subraya que este paradigma permite un desarrollo ágil, altamente escalable y de bajo costo, sin comprometer el rendimiento del sistema. Además, resaltan que el enfoque serverless resulta ideal para contextos donde los patrones de uso son variables, como es el caso de aplicaciones educativas con accesos masivos concentrados en períodos específicos [4].

Ortiz llevó a cabo una propuesta educativa usando tecnología serverless con los servicios de Amazon Web Services (AWS). En lugar de desarrollar una aplicación monolítica, dividió su funcionamiento en partes más pequeñas e independientes, lo que permitió modificar o actualizar ciertos procesos sin afectar el resto del sistema. Gracias a esta estructura, también se logró una respuesta más rápida y una mayor facilidad para adaptarse a diferentes cargas de trabajo. Este trabajo demostró que es posible aplicar este tipo de arquitectura en el ámbito educativo sin necesidad de recurrir a servidores tradicionales o infraestructura compleja [9].

Un ejemplo destacado en América Latina es PreuApp, una iniciativa chilena pensada para apoyar a los estudiantes en su camino hacia la educación superior. Aunque no se trata de una aplicación web ni utiliza arquitectura serverless, ha logrado captar la atención de los usuarios gracias al uso de elementos de gamificación como niveles, monedas virtuales e incluso inteli-

gencia artificial para adaptar el contenido a cada estudiante. Este caso muestra que, cuando se aplican estrategias innovadoras y bien contextualizadas, es posible generar impacto real en la preparación preuniversitaria, incluso sin recurrir a tecnologías de última generación [11].

Desde el punto de vista de evaluación, se ha documentado ampliamente el uso del System Usability Scale (SUS) como herramienta válida y confiable para medir la percepción del usuario respecto a la facilidad de uso de sistemas tecnológicos. Brooke propone esta escala como un instrumento simple pero robusto, capaz de proporcionar datos cuantificables que orienten mejoras en el desarrollo de plataformas educativas [10].

Por otra parte, estudios técnicos como el de Jonas et al. detallan la aplicabilidad de AWS Lambda como servicio central de ejecución en aplicaciones serverless, señalando su eficiencia para disparar funciones en tiempo real, minimizar costos por demanda y mantener una alta disponibilidad del sistema en picos de acceso [5].

El uso de la computación en la nube ofrece beneficios económicos sustanciales, ya que permite un modelo de pago por uso y elimina la necesidad de una gran inversión inicial en infraestructura. Esto es ideal para un proyecto educativo, donde la rentabilidad y accesibilidad son clave. Además, la nube proporciona una escalabilidad bajo demanda para adaptarse a un número variable de usuarios (estudiantes) y ofrece almacenamiento masivo y seguro para todo el contenido educativo, facilitando también la colaboración si se concibe un entorno de aprendizaje compartido [12].

Para una aplicación gamificada que necesita ser dinámica y accesible, la integración con servicios en la nube es esencial. Las plataformas en la nube ofrecen capacidad prácticamente ilimitada de almacenamiento y poder de cómputo, lo que es crucial para manejar la interactividad y los datos generados por la gamificación. Su modelo de pago por lo que se utiliza y la independencia de la ubicación significan que los aspirantes pueden acceder a la preparación desde cualquier dispositivo y lugar con conexión a internet, garantizando eficiencia y amplia disponibilidad [13].

Para una aplicación web educativa y gamificada, una infraestructura subyacente con buen rendimiento es crucial para garantizar una experiencia de usuario fluida y receptiva, evitando demoras que podrían afectar la interacción y la motivación de los estudiantes. Esto implica que la elección del proveedor de la nube y la optimización de sus recursos son determinantes para el éxito de la aplicación [14].

La adopción de una arquitectura serverless constituye un elemento clave dentro de esta propuesta. Este paradigma abstrae la gestión de servidores, permitiendo que los desarrolladores se concentren principalmente en la lógica de la aplicación y en la implementación de los elementos gamificados. Además, ofrece escalado automático ante variaciones en la demanda, como incrementos en la cantidad de usuarios, y reduce costos al utilizar un modelo de pago basado únicamente en los recursos consumidos. En contextos educativos, donde la actividad estudiantil puede fluctuar considerablemente, esta flexibilidad resulta especialmente valiosa para garantizar agilidad, eficiencia y capacidad de adaptación [15].

Finalmente, en lo que respecta al almacenamiento de datos en este tipo de arquitecturas, Bawsay et al. destacan que Amazon DynamoDB se posiciona como una de las bases NoSQL más utilizadas en entornos serverless. Sus características como baja latencia, escalabilidad automática y compatibilidad con AWS Lambda la convierten en una opción ideal para manejar operaciones asincrónicas como el registro y seguimiento de estudiantes en una plataforma educativa [7].

Revisando los antecedentes se observa que la gamificación cuenta con sólidos respaldos tanto en lo técnico como también de lo académico; igual ocurre con el serverless. Combinar ambos planteamientos en una sola aplicación (particularmente como parte de un entorno dedicado a la preparación preuniversitaria) abre la posibilidad de crear opciones más relevantes para el estudiante. Actividades como cuestionarios, ejercicios de asociación y pruebas por niveles incorporadas con utilización de servicios en la nube escalable/accesible es una importante progresión a soluciones educativas más eficientes y mejoradas para nuestros tiempos. No sólo esta integración ayuda a tener experiencias de aprendizaje más dinámicas e interactivas, sino que, además, trae sustentabilidad y flexibilidad tecnológicas, en un contexto donde tanto la demanda

como el modo de aprender están mutando.

*A continuación, la **Tabla II** resume las principales evidencias y su relevancia.*

Tabla II: Antecedentes: síntesis de evidencias y relevancia

Autor y año	**Tema/Enfoque**	**Hallazgo clave**	**Relevancia**
Lampropoulos & Sidiropoulos (2024)	Gamificación en educación	Los entornos gamificados favorecen mejores calificaciones y duplican el rendimiento académico destacado frente a modelos tradicionales y completamente virtuales.	Respalda el uso de estrategias de gamificación.
Li, Ma & Shi (2023)	Metaanálisis (41 estudios)	Se identificó un efecto positivo sostenido del uso de elementos gamificados, con un tamaño del efecto elevado (0.82).	Fundamenta el uso de retos, puntos y niveles.
Hassan, Barakat & Sarhan (2021)	Revisión de serverless	La arquitectura serverless permite un desarrollo ágil, escalable y con costos reducidos, especialmente en contextos de demanda variable.	Sustenta la adopción de arquitectura serverless.
Ortiz (2019)	Caso educativo en AWS	El uso de funciones independientes mejora la respuesta del sistema y facilita modificaciones sin afectar la aplicación completa.	Evidencia la viabilidad técnica de la propuesta.
PreuApp (2025)	Caso latinoamericano	La incorporación de niveles, monedas virtuales e IA generó resultados positivos en la experiencia educativa.	Demuestra el potencial de la estrategia gamificada.

Continúa en la siguiente página

Tabla II (continuación). Antecedentes: síntesis de evidencias y relevancia

Autor y año	Tema/Enfoque	Hallazgo clave	Relevancia
Brooke (1996)	System Usability Scale	La escala SUS ofrece una forma simple y confiable de evaluar la usabilidad.	Define el instrumento de evaluación.
Jonas et al. (2019)	AWS Lambda (FaaS)	AWS Lambda permite ejecución en tiempo real, costos según demanda y alta disponibilidad ante picos de uso.	Respalda la elección del modelo FaaS.
Bello (2021)	Beneficios de cloud	La computación en la nube proporciona escalabilidad, almacenamiento seguro y pago por uso.	Justifica el empleo de servicios cloud.
Pierleoni et al. (2019)	Cloud para apps dinámicas	La capacidad de procesamiento y el acceso ubicuo de la nube favorecen aplicaciones más interactivas y dinámicas.	Sustenta la integración de tecnologías cloud.
Zbořil & Svata (2025)	Rendimiento de nube	Un alto rendimiento en la nube resulta esencial para garantizar una experiencia de usuario fluida.	Define criterios para seleccionar el proveedor tecnológico.
Toosi et al. (2025)	Justificación serverless	El escalado automático y el pago por uso aportan flexibilidad y eficiencia operativa.	Se adapta a variaciones en la demanda estudiantil.

Continúa en la siguiente página

Tabla II (continuación). Antecedentes: síntesis de evidencias y relevancia

Autor y año	Tema/Enfoque	Hallazgo clave	Relevancia
Bawsay et al. (2025)	DynamoDB en serverless	DynamoDB ofrece baja latencia, autoescalado e integración eficiente con AWS Lambda.	Presenta una alternativa adecuada para la gestión de datos.

Fuentes: [1], [4], [5], [7], [8], [9], [10], [11], [12], [13], [14], [15]

2.2 Educación Preuniversitaria

2.2.1 Conceptualización de la Educación Preuniversitaria

La educación preuniversitaria (PUE, por sus siglas en inglés) es un programa diseñado para ofrecer servicios de apoyo personalizados y enseñanza de cursos preuniversitarios, con el propósito específico de preparar a los estudiantes que no están suficientemente preparados para tener éxito en el nivel universitario. Se le conoce con diversas denominaciones como "educación precolegial", "habilidades básicas postsecundarias", "cursos correctivos", "remediación", "educación de desarrollo" o "programa alternativo". La prevalencia de la educación preuniversitaria está bien establecida en la literatura sobre educación superior, con una participación significativa de estudiantes en Estados Unidos y países subsaharianos como Nigeria. Es vista como una responsabilidad colegiada que las instituciones de educación superior deberían integrar en su sistema [16].

El objetivo general de la educación profesional, que incluye elementos preuniversitarios, es el desarrollo integral de la persona como individuo, cultivando sus talentos, capacidades intelectuales, creativas y físicas. Esto implica la formación de valores y competencias necesarias para una autorrealización exitosa, la educación de ciudadanos responsables capaces de tomar

decisiones sociales conscientes y dirigir sus actividades en beneficio de otras personas y la sociedad [17].

En seguida aparece una síntesis conceptual acerca de la educación preuniversitaria en la Tabla III.

Tabla III: Educación preuniversitaria: conceptos y alcance

Aspecto	**Descripción**
Definición (PUE)	Programa de acompañamiento personalizado y cursos preuniversitarios orientados a preparar estudiantes con brechas académicas para el entorno universitario.
Denominaciones	Educación precolegial, habilidades básicas postsecundarias, cursos correctivos, remediación, educación de desarrollo y programas alternativos.
Prevalencia	Participación considerable en EE. UU. y en países del África subsahariana, como Nigeria.
Responsabilidad institucional	Función compartida que las instituciones de educación superior deben integrar dentro de su sistema formativo.

Continúa en la siguiente página

Tabla III (continuación). Educación preuniversitaria: conceptos y alcance

Aspecto	**Descripción**
Objetivo general	Promover el desarrollo integral mediante el fortalecimiento de capacidades intelectuales, creativas y físicas, junto con la formación de valores y competencias para la autorrealización y la ciudadanía responsable.

Fuentes: [16], [17]

A continuación, se resume una selección de estos elementos en la Tabla IV.

Tabla IV: Elementos fundamentales en la educación preuniversitaria

Elemento	**Idea clave**
Autodesarrollo de la personalidad	Proceso continuo de construcción de identidad que orienta el proyecto de vida y las decisiones vocacionales, fortaleciendo la formación académica y personal.
Actividades extracurriculares	Espacios organizados que favorecen el desarrollo profesional mediante enfoques tradicionales y no tradicionales, además de especializaciones complementarias.

Continúa en la siguiente página

Tabla IV (continuación). Elementos fundamentales en la educación preuniversitaria

Elemento	**Idea clave**
Inteligencia emocional (IE)	Conjunto de habilidades para reconocer, expresar y regular emociones, favoreciendo el funcionamiento cognitivo y la resolución de problemas.
Aprendizaje autodirigido (SDL)	Promueve autonomía, responsabilidad y crecimiento personal; además, actúa como medio a través del cual la IE influye en el rendimiento académico.
Integración de e -learning	Facilita personalización, acceso a recursos y colaboración; los LMS integran aprendizaje, evaluación y retroalimentación continua.
Gamificación y microaprendizaje	El uso de dinámicas de juego y microcontenidos (<10 min) incrementa el compromiso, la retención y la participación del estudiante.
Interacción	La interacción formal e informal entre estudiantes y docentes favorece la integración y el éxito académico, especialmente en programas de transición.

Continúa en la siguiente página

Tabla IV (continuación). Elementos fundamentales en la educación preuniversitaria

Elemento	Idea clave
Sentido de pertenencia	Percepción de apoyo, integración y reconocimiento que contribuye a reducir el abandono durante la transición al entorno universitario.

Elaboración propia. Fuentes: [17], [18], [19]

2.2.2 Beneficios de la Educación Preuniversitaria y sus Componentes

La inversión en educación preuniversitaria y sus elementos clave se traduce en beneficios sustanciales para el estudiante y el sistema educativo.

Mejora del rendimiento académico. Los programas de transición ayudan a darle una ventaja: incrementa las calificaciones del primer curso, y eleva el GPA del primer año [19]. La IE (interpersonal e intrapersonal) y el aprendizaje dirigido por sí mismo predicen positivamente al rendimiento [18]. Se revela que una combinación de e-learning y gamificación tiene un gran potencial [16].

Desarrollo integral de la persona. Se fomenta el desarrollo multifacético (talentos, capacidades intelectuales, creativas y físicas), la formación de valores y competencias para la autorrealización, así como la construcción de identidad profesional y autoeficacia [17].

Preparación para resolver problemas sociales y profesionales. Se potencia la autodeterminación en el entorno educativo, el pensamiento crítico y la autonomía para resolver tareas profesionales [17].

Mejora de competencias profesionales. Las actividades extracurriculares contribuyen al desa-

rrollo de distintos perfiles profesionales, como consultor, analista, gestor o facilitador, además de fortalecer habilidades y competencias aplicadas. De igual forma, el e-learning facilita la adquisición de nuevos conocimientos y su integración con aprendizajes previos [17], [18].

Autorregulación y autoeficacia. El autodesarrollo favorece la autorregulación y fortalece la percepción que el estudiante tiene sobre sus propias capacidades, mientras que el aprendizaje autodirigido impulsa la motivación intrínseca y la confianza en el proceso formativo [16], [17].

Compromiso y persistencia. La inteligencia emocional contribuye a la perseverancia académica, y un sólido sentido de pertenencia incrementa la motivación del estudiante y disminuye el riesgo de abandono [16], [19].

Adaptación a demandas del siglo XXI. Las TIC, la inteligencia artificial y el IoT aportan mayor flexibilidad y acceso a recursos educativos, además de fortalecer la preparación para los exámenes de ingreso y transformar las formas de colaboración y acceso a la información [18].

Gestión del estrés y adaptabilidad. La IE intrapersonal es clave para afrontar el estrés académico; se recomiendan intervenciones centradas en IE dentro de la PUE [16].

A continuación, se presenta la síntesis de estos beneficios en la Tabla V.

Tabla V: Beneficios de la educación preuniversitaria y sus componentes

Beneficio	**Evidencia/idea clave**
Rendimiento académico	Programas de transición elevan el GPA del primer año; IE (intra/interpersonal) y SDL predicen rendimiento; e-learning y gamificación apoyan mejoras [16], [18], [19].

Continúa en la siguiente página

Tabla V (continuación). Beneficios de la educación preuniversitaria y sus componentes

Beneficio	**Evidencia/idea clave**
Desarrollo integral	Formación de valores, competencias y autoeficacia; construcción de identidad profesional [17].
Resolución de problemas	Fomenta autodeterminación, pensamiento crítico y autonomía para tareas profesionales [17].
Competencias profesionales	Actividades extracurriculares amplían roles y perfeccionan cualidades; e -learning facilita conexiones cognitivas [17], [18].
Autorregulación y autoeficacia	El autodesarrollo fortalece la autorregulación; el SDL impulsa motivación intrínseca y confianza [16], [17].
Compromiso y persistencia	La IE apoya la perseverancia; el sentido de pertenencia incrementa motivación y reduce abandono [16], [19].
Adaptación siglo XXI	TIC, IA e IoT favorecen flexibilidad, acceso y preparación para exámenes [18].

Continúa en la siguiente página

Tabla V (continuación). Beneficios de la educación preuniversitaria y sus componentes

Beneficio	Evidencia/idea clave
Gestión del estrés	La IE intrapersonal mejora el afrontamiento del estrés; se recomiendan intervenciones específicas [16].

2.3 Tecnologías Educativas

2.3.1 Conceptualización de la Tecnología

El término "tecnología" es polisémico en el ámbito educativo. Etimológicamente proviene de téchne (técnica, oficio) y lógos (conocimiento), como conjunto de saberes que permiten aplicar una técnica u oficio. Es reconocido como un sistema socio-técnico compuesto por hardware, reglas y los sistemas; una mezcla de visible e invisible. Se distingue así entre técnica (prácticas artesanales precientíficas) y tecnología (técnica industrial ligada al saber científico). La tecnología es también .[el] discurso sobre la técnica". Se puede ver como reorganización física de material de acuerdo con las ideas/teorías humanas(encarnación material de teoríaes, un punto de contacto la mente–el mundo),y como un sistema de acción humana orientado a convertir objetos para crear valor, en donde las personas tienen el papel de actores principales. El papel de la tecnología es el de detonante transformador del mundo; crea tecnosferas y afecta a cualquier esfera de la acción humana. La UNESCO y otros organismos consideran la tecnología como un saber práctico y un proceso creativo dirigido al uso de recursos, herramientas y sistemas con el fin de solucionar problemas. Se entiende, además, desde una perspectiva más amplia como un conjunto organizado de conocimientos y procesos afines para la producción de bienes y servicios. Integra dimensiones científico-técnicas, socioeconómicas y culturales. En este sentido, la tecnología no solo constituye un conjunto de herramientas, sino también un elemento cultural

que influye en las interacciones y comportamientos sociales. Su alcance incluye tanto artefactos tangibles e intangibles como las personas, infraestructuras y procesos necesarios para diseñarlos, operarlos y mantenerlos [20].

A continuación, se presenta una síntesis de estas conceptualizaciones en la Tabla VI.

Tabla VI: Síntesis de conceptualizaciones de tecnología

Eje conceptual	**Idea principal**
Origen etimológico	Proviene de *téchne* y *lógos*, términos asociados al conocimiento aplicado a la técnica y al oficio.
Sistema socio–técnico	Integra hardware, normas y sistemas, articulando componentes visibles e invisibles dentro de un mismo entorno tecnológico.
Técnica vs. tecnología	La técnica se relaciona con prácticas artesanales precientíficas, mientras que la tecnología se vincula al conocimiento científico y al análisis de la técnica.
Encarnación de teorías	La tecnología reorganiza el entorno físico a partir de ideas y conocimientos, actuando como punto de conexión entre pensamiento y realidad material.
Acciones humanas	Conjunto de acciones intencionales orientadas a transformar objetos, procesos y recursos para generar valor.

Continúa en la siguiente página

Tabla VI (continuación). Síntesis de conceptualizaciones de tecnología

Eje conceptual	Idea principal
Transformación del mundo	Impulsa el desarrollo de la tecnosfera y atraviesa prácticamente todas las actividades humanas.
Definición UNESCO	Se entiende como un saber práctico y un proceso creativo destinado a resolver problemas y mejorar la condición humana.
Visión amplia	Comprende conocimientos y procesos utilizados para producir bienes y servicios, incorporando dimensiones sociales, económicas y culturales.
Artefacto cultural	Influye en las interacciones y conductas humanas, incluyendo artefactos tangibles, intangibles y el entorno humano que los sostiene.

Elaboración propia. Fuente: [20]

2.3.2 Conceptualización de la Tecnología Educativa

La tecnología educativa puede entenderse, por un lado, como el uso instrumental de dispositivos y recursos tecnológicos con fines pedagógicos y, por otro, como un *proceso* sistemático de aplicación de tecnología para mejorar la calidad educativa. Es un campo dinámico que integra investigación y práctica, abarcando herramientas instruccionales, recursos y enfoques basados en discurso tecnológico para estructurar las dinámicas de enseñanza–aprendizaje. Tradicionalmente se asoció con computadoras, Internet y móviles, junto con temas como diseño instruccional,

aprendizaje móvil, redes sociales y recursos educativos abiertos (REA). La AECT la concibe como la aplicación de tecnología para mejorar la eficiencia y efectividad educativa mediante funcionalidades nuevas o mejoradas [20], [21], [22], [23].

A continuación, se sintetizan estas perspectivas en la Tabla VII.

Tabla VII: Síntesis de conceptualizaciones de tecnología educativa

Perspectiva	**Definición/alcance**
Uso instrumental	Empleo de dispositivos y recursos tecnológicos con propósitos educativos y formativos.
Proceso sistemático	Aplicación organizada y sistemática de tecnologías orientadas a mejorar la calidad del proceso educativo.
Campo híbrido (investigación–práctica)	Conjunto de herramientas, recursos y enfoques que permiten analizar, estructurar y fortalecer las dinámicas de enseñanza y aprendizaje.
Enfoque tradicional	Perspectiva centrada en hardware, redes y temas como diseño instruccional, aprendizaje móvil, redes sociales y recursos educativos abiertos.
Definición AECT	Aplicación de tecnologías orientadas a mejorar la eficiencia y efectividad educativa mediante funcionalidades nuevas o optimizadas.

Elaboración propia. Fuentes: [20], [21], [22], [23]

2.4 Gamificación en el Entorno Educativo

2.4.1 Definición de Gamificación

La gamificación busca transformar tareas rutinarias en experiencias lúdicas mediante el uso de elementos y mecánicas de juego para motivar e involucrar a los estudiantes [24]. En el ámbito académico se la define como la incorporación de elementos de diseño, mecánicas, estética y pensamiento de juego en contextos no lúdicos para promover la participación y el aprendizaje [25]. Deterding et al. la entienden como el uso de componentes de juego en contextos que no son juegos; Kapp et al. enfatizan su propósito de involucrar, motivar, apoyar el aprendizaje y resolver problemas. Seaborn y Fels sintetizan estas perspectivas como el uso intencional de elementos de juego para generar una experiencia lúdica en tareas no lúdicas [26]. Más allá de la educación, también se aplica en negocios y salud para capacitación, *engagement* y cambio de comportamiento, aprovechando el disfrute que evoca [27]. Aunque el término es reciente, sus mecánicas (recompensas, lealtad, calificaciones) son previas y hoy se expanden por el abaratamiento tecnológico y la prevalencia de los videojuegos [26]. Se relaciona con juegos serios y aprendizaje basado en juegos: los primeros son juegos completos con contenido educativo; la gamificación reutiliza componentes de juego en entornos reales, orientando el aprendizaje de forma indirecta al modificar motivación y conducta [25], [26].

A continuación, se sintetizan las definiciones y el alcance en la Tabla VIII.

Tabla VIII: Definiciones y alcance de la gamificación

Enfoque	**Definición/alcance**
Funcional	Uso de elementos y mecánicas de juego orientados a incrementar la motivación y participación de los estudiantes.

Continúa en la siguiente página

Tabla VIII (continuación). Definiciones y alcance de la gamificación

Enfoque	Definición/alcance
Académico	Aplicación de elementos, mecánicas, estética y pensamiento de juego en contextos no lúdicos para fortalecer el aprendizaje.
Síntesis conceptual	Incorporar intencionadamente dinámicas de juego para obtener experiencias más interesantes dentro de un videojuego, conocidas como actividades no lúdicas.
Ámbitos de aplicación	Se puede aplicar en áreas como la educación, los negocios y la salud. Favorece los procesos de capacitación.
Relación con afines	Es diferente de Juego Serio e Inducción Basada en Juegos: Aprender por medio de Motivación y Conducta Indirectas.

Elaboración propia. Fuentes: [24], [25], [26], [27]

2.4.2 Elementos de Gamificación

En entornos educativos, los elementos de diseño de juego más utilizados incluyen puntos, tablas de clasificación, distintivos, niveles, desafíos o misiones, barras de progreso, avatares y narrativa. La selección de estos componentes debe responder a necesidades psicológicas como la competencia, la autonomía y la interacción social, además de considerar las características del estudiantado. Las implementaciones realizadas sin una planificación adecuada suelen reducir la efectividad y la posibilidad de replicar resultados en otros contextos. En cambio, cuando la gamificación se fundamenta teóricamente y se integra con estrategias pedagógicas, puede

fortalecer significativamente la motivación y el aprendizaje. En áreas como las ciencias, la gamificación digital también ha mostrado capacidad para disminuir emociones negativas y facilitar la comprensión de conceptos complejos [25], [26].

A continuación, se presenta la Tabla IX con los elementos principales y su propósito educativo.

Tabla IX: Elementos de gamificación y propósito educativo

Elemento	**Propósito/efecto esperado**
Puntos, distintivos, logros	Proporcionan retroalimentación inmediata, fortalecen la percepción de competencia y favorecen el establecimiento de metas.
Tablas de clasificación	Incentivan la comparación social y la competencia, aunque un uso excesivo puede afectar el disfrute de la experiencia.
Niveles, barras de progreso	Permiten visualizar el avance del estudiante y facilitan la organización gradual de objetivos.
Desafíos, misiones	Plantean metas concretas que ayudan a mantener el interés y la atención durante el aprendizaje.
Avatares	Favorecen la proyección de identidad y el modelado de conductas deseadas dentro del entorno educativo.

Continúa en la siguiente página

Tabla IX (continuación). Elementos de gamificación y propósito educativo

Elemento	Propósito/efecto esperado
Narrativa (storytelling)	Aporta sentido y continuidad a las actividades, apoyando el logro de objetivos de aprendizaje.
Diseño fundamentado	La selección de elementos debe responder a necesidades psicológicas y características del contexto, evitando implementaciones improvisadas.
Entornos digitales	Contribuyen a disminuir emociones negativas en áreas como ciencias y favorecen el uso de plataformas educativas como Moodle o Kahoot!.

Elaboración propia. Fuentes: [25], [26]

2.4.3 Beneficios de la Gamificación en el Aprendizaje

La evidencia empírica reporta mejoras en motivación, compromiso, comportamiento y resultados de aprendizaje cuando se incorporan elementos de juego en experiencias educativas. En comparaciones con métodos tradicionales se observan desempeños superiores; por ejemplo, en biología celular se reportó una mejora cuantitativa destacada. La motivación extrínseca inicial (recompensas) puede progresar hacia motivación intrínseca en el marco de la Teoría de la Autodeterminación (SDT), mientras que modelos como ARCS explican cómo la satisfacción impulsa el compromiso sostenido. También se documentan aumentos en autoeficacia, disfrute y actitudes positivas, así como el desarrollo de habilidades transversales y posibilidades de

personalización y analítica educativa [24], [25], [26], [27], [28].

A continuación, la Tabla X sintetiza los principales beneficios y hallazgos asociados.

Tabla X: Beneficios de la gamificación en educación

Categoría	**Hallazgos clave**
Motivación y compromiso	Los entornos gamificados generan experiencias más atractivas y fortalecen la motivación extrínseca e intrínseca; además, la satisfacción influye directamente en el compromiso del estudiante (ARCS + SDT).
Rendimiento académico	Se reportan mejores resultados en comparación con metodologías tradicionales, incluyendo mejoras cuantificables en asignaturas específicas.
Autoeficacia y confianza	Se observan incrementos significativos en la confianza del estudiante, reforzados mediante logros, recompensas y distintivos.
Afecto y disfrute	La gamificación favorece mayores niveles de disfrute, entusiasmo y actitudes positivas hacia el aprendizaje.

Continúa en la siguiente página

Tabla X (continuación). Beneficios de la gamificación en educación

Categoría	Hallazgos clave
Habilidades transversales	Las dinámicas lúdicas fortalecen habilidades como resolución de problemas, cooperación y comunicación.
Personalización y autonomía	Permite ajustar el nivel de desafío, ofrecer rutas de aprendizaje diferenciadas y promover mayor autonomía en la progresión.
Recolección de datos	Facilita la trazabilidad de interacciones y el análisis educativo en tiempo real para procesos de retroalimentación y mejora continua.
Cambio de comportamiento	También se aplica en salud y otros ámbitos para incentivar hábitos deseados y mejorar la adherencia a determinadas actividades.

Elaboración propia. Fuentes: [24], [25], [26], [27], [28]

2.5 Aplicaciones Web Educativas

2.5.1 Definición de Aplicación Web

Una aplicación web es un sistema de software accesible mediante navegador que opera típicamente bajo una arquitectura cliente–servidor. El cliente utiliza HTML, CSS y JavaScript,

mientras que el servidor implementa la lógica, el acceso a datos y la integración con bases de datos mediante lenguajes como PHP o Node.js y *frameworks* como Flask, sobre servidores como Apache o Tomcat [29]. Esta arquitectura habilita la recolección, procesamiento y visualización dinámica de datos, con bajo requisito de recursos en el equipo del usuario y accesibilidad ubicua. En educación, favorece la participación activa y la motivación estudiantil [30].

La siguiente tabla XI resume la arquitectura y las ventajas de una aplicación Web en el contexto educativo.

Tabla XI: Aplicación web: estructura y ventajas en educación

Aspecto	**Descripción**
Arquitectura	Modelo cliente–servidor en el que el cliente utiliza HTML, CSS y JavaScript, mientras el servidor gestiona la lógica de negocio y la integración con bases de datos.
Tecnologías del servidor	Uso de lenguajes y *frameworks* como PHP, Node.js y Flask, junto con servidores como Apache o Tomcat.
Ventajas	Requiere pocos recursos, permite acceso desde el navegador y facilita la interacción dinámica con la información.
Impacto educativo	Favorece la participación, incrementa la motivación y promueve experiencias de aprendizaje más activas e interactivas.

Elaboración propia. Fuentes: [29], [30]

2.5.2 Aplicaciones Web Educativas: Concepto General

Las aplicaciones web educativas son programas diseñados para apoyar la enseñanza y el aprendizaje, transformando metodologías tradicionales en experiencias eficientes, atractivas y personalizadas. Cubren planificación curricular, entrega de contenidos y evaluación a lo largo de distintos niveles educativos, promoviendo la autonomía del estudiante y entornos flexibles que trascienden restricciones de tiempo y espacio. Además, posibilitan analítica del comportamiento y del rendimiento para la mejora continua [30], [31].

La Tabla XII a continuación resume los siguientes objetivos y aportes principales.

Tabla XII: Aplicaciones web educativas: objetivos y aportes

Dimensión	**Descripción**
Ámbito	Dan apoyo al contenido, los procesos curriculares y las actividades de evaluación, en todos los niveles educativos.
Experiencia	Favorece un aprendizaje más dinámico (hacer). Aprendizaje personalizado y activo que propicia la participación.
Flexibilidad	Permite superar restricciones con respecto a espacio y tiempo a través del acceso ubicuo por medio de un browser web.
Analítica	Facilita el monitoreo del desempeño y conducta estudiantil para un proceso continuo de retroalimentación y mejoramiento educativo.

Elaboración propia. Fuentes: [30], [31]

2.5.3 Tipos de Aplicaciones Web Educativas

El ecosistema de software educativo basado en la web abarca desde sistemas integrales de gestión del aprendizaje hasta herramientas especializadas para evaluación, tutoría inteligente, simulación, gamificación y comunicación. Esta diversidad permite cubrir necesidades de diseño curricular, entrega de contenidos, retroalimentación y colaboración en múltiples contextos [29], [31], [32].

La Tabla XIII sintetiza los tipos más comunes y su propósito principal.

Tabla XIII: Tipos comunes de aplicaciones web educativas

Tipo	**Propósito y ejemplos**
LMS	Permiten gestionar cursos, inscripciones, materiales, evaluaciones y seguimiento académico; por ejemplo, Moodle y Google Classroom.
Evaluación y retroalimentación	Facilitan exámenes, calificaciones y *feedback* personalizado mediante bancos de preguntas y cuestionarios gamificados.
Tutoría inteligente (ITS)	Sistemas apoyados en inteligencia artificial orientados al aprendizaje individualizado y la asistencia por asignaturas.
Autoría de contenido	Herramientas destinadas a la creación de materiales multimedia y objetos de aprendizaje reutilizables.

Continúa en la siguiente página

Tabla XIII (continuación). Tipos comunes de aplicaciones web educativas

Tipo	Propósito y ejemplos
Simulación	Entornos y modelos interactivos utilizados en ciencias y formación técnica para representar escenarios reales.
Gamificación y juegos	Incorporan dinámicas de juego para fortalecer la motivación, la práctica progresiva y la evaluación formativa; por ejemplo, Kahoot! y Quizizz.
Motores de recomendación	Personalizan recursos y rutas de aprendizaje mediante analítica y técnicas de filtrado.
Editores matemáticos	Permiten la transcripción y representación de expresiones o figuras matemáticas en tiempo real, como ocurre en plataformas LaTeX online.
Comunicación síncrona	Facilitan videoconferencias y chats para clases en vivo mediante herramientas como Meet, Teams, Zoom y BigBlueButton.

Elaboración propia. Fuentes: [29], [31], [32]

2.6 Tecnologías para el Desarrollo de Aplicaciones Web

El desarrollo web contemporáneo se sustenta en estándares consolidados del lado del cliente (HTML, CSS y JavaScript) y en runtimes/lenguajes de servidor ampliamente adoptados (p. ej., Node.js, Python, PHP, C++). Para cargas intensivas en el navegador, WebAssembly permite ejecutar módulos cercanos al rendimiento nativo, mientras que APIs del navegador (Canvas/WebGL, Web Audio, Storage) habilitan experiencias ricas y persistencia local [33].

Estas tecnologías base y su rol principal se resumen en la tabla XIV

Tabla XIV: Tecnologías base para aplicaciones web

Tecnología	**Rol principal**
HTML	Describe estructura y significado del contenido de la Web como soporte de la accesibilidad y el SEO.
CSS	Diseño de apariencia; controla su distribución, tipografía etc., Flexbox, Grid entre otros. Fácil para diseñar con responsive design.
JavaScript	JS es un manejador de la lógica que interactúa con el usuario, la manipulación del DOM, la comunicacion por Fetch... Además el acceso a APIs web (Storage, WEB GL).
WebAssembly	A diferencia del Javascript actual, permite correr módulos con alto performance en el lado cliente compilandolos. (desde C++).

Continúa en la siguiente página

Tabla XIV (continuación). Tecnologías base para aplicaciones web

Tecnología	Rol principal
Lenguajes/runtimes backend	Se encargan de implementar los servicios, APIs y el acceso a datos con tecnología Node.Js , Python, Php, etc.

Elaboración propia. Fuente: [33]

2.6.1 Arquitectura de Aplicaciones Web

Las aplicaciones web modernas suelen separar la interfaz de usuario (frontend) de la capa de servidor (backend) para facilitar la escalabilidad, el mantenimiento y la distribución de responsabilidades. En las *single-page applications* (SPA), la interfaz se hidrata y mantiene su estado directamente en el cliente, permitiendo interacciones más dinámicas y fluidas. Por otro lado, las *multi-page applications* (MPA) delegan una mayor parte del renderizado y del manejo del estado al servidor entre solicitudes. De forma paralela, los enfoques dirigidos por modelos (MDA) organizan el desarrollo mediante distintos niveles de abstracción, como CIM, PIM y PSM, apoyándose en transformaciones formales entre modelos. Asimismo, han surgido propuestas como la *resumabilidad*, orientadas a reanudar la interacción de la aplicación sin necesidad de ejecutar nuevamente toda la lógica del sistema. En entornos cloud, el uso de contenedores y microservicios mejora la portabilidad, el despliegue y la flexibilidad operativa de las aplicaciones web [34].

La Tabla XV sintetiza los modelos y patrones arquitectónicos más relevantes para este proyecto.

Tabla XV: Modelos y patrones arquitectónicos

Modelo	**Descripción**
SPA	Interfaz reactiva construida sobre una sola página, con hidratación del lado del cliente, enrutamiento interno y manejo de estado en memoria.
MPA	Modelo basado en renderizado del lado del servidor y navegación entre múltiples páginas; favorece estabilidad inicial y posicionamiento SEO.
MDA (CIM/PIM/PSM)	Enfoque de desarrollo dirigido por modelos que utiliza transformaciones formales entre distintos niveles de abstracción.
Resumabilidad	Permite reanudar estados y eventos sin volver a ejecutar toda la aplicación, reduciendo el costo de hidratación.
Contenedores	Facilitan el empaquetado consistente de servicios y dependencias, mejorando la portabilidad y el despliegue escalable.

Elaboración propia. Fuente: [34]

2.6.2 Herramientas y Programas Utilizados

Los navegadores web proporcionan un entorno seguro de ejecución mediante mecanismos como el *sandbox*, las políticas de mismo origen y la gestión explícita de permisos. Por su parte, los servidores web se encargan de distribuir recursos y administrar el acceso a archivos y bases de datos. A esto se suman los simuladores de red, que permiten validar topologías, automatizar configuraciones y comprobar la conectividad entre componentes. En conjunto, estas herramientas sostienen los procesos de desarrollo, pruebas y operación de aplicaciones web educativas [33], [35].

La Tabla XVI resume las herramientas de plataforma y soporte consideradas en este trabajo.

Tabla XVI: Herramientas de plataforma y soporte

Categoría	**Función**
Navegadores	Permiten la ejecución segura mediante *sandbox*, acceso a APIs Web como Storage, Fetch, WebGL y Audio, además de la gestión de permisos y políticas de mismo origen.
Servidores web	Gestionan contenidos estáticos y dinámicos, terminación TLS y comunicación con servicios backend y bases de datos mediante CGI, FastCGI o WSGI.
Gestores de aplicaciones	Facilitan el empaquetado y ejecución de aplicaciones web con integración al sistema operativo para distribución y actualización.

Continúa en la siguiente página

Tabla XVI (continuación). Herramientas de plataforma y soporte

Categoría	**Función**
Simuladores de red	Permiten diseñar, probar y automatizar topologías, validando conectividad, protocolos y *scripts* de operación.

Elaboración propia. Fuentes: [33], [35]

2.6.3 Entornos de Desarrollo y Bibliotecas

Los entornos de modelado, como EMF, y las herramientas de transformación, como QVTo y Acceleo, facilitan la aplicación de enfoques dirigidos por modelos durante el desarrollo de software. En la capa de presentación, motores de plantillas como Apache Velocity apoyan la implementación de arquitecturas MVC y la generación dinámica de vistas. Para la construcción de servicios web ligeros se emplea Flask junto con su ecosistema en Python. Además, plataformas como GitHub y SourceForge, junto con tecnologías de distribución descentralizada como IPFS, favorecen la colaboración, el control de versiones y la distribución de artefactos de software [33], [35], [36].

La Tabla XVII resume bibliotecas y herramientas consideradas y su papel primario en la librería del proyecto.

Tabla XVII: Entornos y bibliotecas destacadas

Herramienta	**Uso principal**
EMF	Permite modelar metamodelos y generar artefactos para acelerar el desarrollo basado en modelos.
QVTo / Acceleo	Realizan transformaciones modelo–modelo (M2M) y modelo–texto (M2T) para generar código y artefactos de despliegue.
Apache Velocity	Motor de plantillas utilizado en arquitecturas MVC para la generación dinámica de vistas y contenido.
Flask y ecosistema	Facilitan la construcción de microservicios web en Python, incluyendo formularios, vistas, autenticación e integración con ORM y *blueprints*.
GitHub / SourceForge / IPFS	Favorecen la colaboración, el control de versiones y la distribución centralizada o descentralizada de artefactos de software.

Elaboración propia. Fuentes: [33], [35], [36]

2.6.4 Bases de Datos en Aplicaciones Web

Las bases de datos sustentan el backend de aplicaciones intensivas en datos. En entornos web, los RDBMS siguen siendo predominantes por su solidez transaccional y madurez de ecosistema; su integración mediante mapeo objeto–relacional (ORM) mejora la productividad y la consistencia. La generación de código para operaciones CRUD reduce trabajo repetitivo, mientras que las migraciones controladas y la ingeniería inversa aceleran la evolución del modelo lógico y físico [36], [37].

La Tabla XVIII resume las prácticas de persistencia y automatización consideradas.

Tabla XVIII: Prácticas de persistencia y automatización

Aspecto	**Descripción**	**Fuente**
RDBMS y ORM	Integración de sistemas gestores de bases de datos relacionales con ORM para mejorar la productividad, mantener la consistencia y facilitar los procesos de prueba.	[36]
CRUD generado	Uso de plantillas y mecanismos de *scaffolding* que disminuyen el trabajo repetitivo en operaciones de creación, eliminación, modificación y consulta.	[36]
Migraciones / generación de esquemas	Implementación de scripts versionados que permiten crear, modificar y revertir esquemas manteniendo trazabilidad en los cambios.	[36]
Ingeniería inversa	Obtención de modelos y entidades a partir de bases de datos existentes para agilizar procesos de actualización y modernización.	[36]
Rol en la web	Componente central del backend en aplicaciones orientadas a datos, ofreciendo soporte para consistencia y consultas complejas.	[37]

Elaboración propia. Fuentes: [36], [37]

2.6.5 Frameworks para el Desarrollo Web

Los frameworks y los runtimes marcan el ritmo del ciclo de vida de las aplicaciones: Renderizado, Enrutamiento, Seguridad, Acceso a Datos... En la interfaz frontal y trasera existen un p. ligero. Ej.: Flask en python) con motores de plantilla p. ej. ej., Apache Velocity) en arquitecturas MVC. Con aplicación de escritorio basado en tecnología Web, los contenedores tales como NW.js o Electron son capaces de empaquetar Chromium con Node.js para poder reutilizar el código [33], [35], [36].

La Tabla XIX sintetiza las herramientas y su rol en la solución.

Tabla XIX: Frameworks y *runtimes* relevantes

Herramienta	**Descripción y uso**
Flask	Microframework de Python orientado al desarrollo de servicios web, formularios y vistas, con un despliegue flexible y escalable.
Apache Velocity	Motor de plantillas utilizado en la capa de presentación bajo el patrón arquitectónico MVC.
Frameworks multiplataforma	Permiten reutilizar una misma base de código en distintos sistemas operativos y entornos de ejecución.
NW.js	Entorno basado en Chromium y Node.js que permite empaquetar aplicaciones web como aplicaciones de escritorio, aunque con mayor tamaño binario.

Continúa en la siguiente página

Tabla XIX (continuación). Frameworks y *runtimes* relevantes

Herramienta	Descripción y uso
Electron JS	Framework similar a NW.js, empleado en aplicaciones como VS Code, que integra Chromium y Node.js para el empaquetado de aplicaciones multiplataforma.

Elaboración propia. Fuentes: [33], [35], [36]

Generación de código. La generación de artefactos mediante plantillas permite agilizar el desarrollo al automatizar tareas repetitivas, como las operaciones CRUD. Dentro de este enfoque, propuestas dirigidas por modelos como E-MDAV y su predecesor MDA-VIEW facilitan la automatización de transformaciones y la preparación de paquetes de despliegue, incluyendo entornos orientados a la nube [36].

La Tabla XX resume las estrategias empleadas.

Tabla XX: Estrategias de generación de código

Estrategia	Aplicación principal
Scaffolding	Permite generar operaciones CRUD y vistas a partir de plantillas y especificaciones predefinidas.
E-MDAV	Automatiza transformaciones basadas en MDA y produce artefactos preparados para entornos cloud.
MDA-VIEW	Facilita la generación de interfaces web a partir de bases de datos ya existentes.

Elaboración propia. Fuente: [36]

2.6.6 Conceptos Adicionales Clave

La ingeniería web aporta métodos y herramientas para desarrollar sistemas orientados a la web, gestionando navegación, seguridad y cambios rápidos, con fundamento en enfoques dirigidos por modelos (MDE). En metodologías, los enfoques tradicionales priorizan planificación rigurosa y documentación; los ágiles favorecen la adaptación a requisitos cambiantes; y los híbridos combinan prácticas según el contexto. El consumo de APIs web garantiza interoperabilidad; además, existen propuestas para descubrimiento y recomendación automática de APIs compatibles. La automatización de procesos (RPA), junto con chatbots con IA y automatización de redes, optimiza flujos y reduce errores operativos [37], [38], [39].

La Tabla XXI sintetiza estos conceptos y su aporte operativo.

Tabla XXI: Conceptos adicionales: síntesis operativa

Concepto	**Enfoque / aporte**
Ingeniería Web	Conjunto de métodos y herramientas orientados al desarrollo de sistemas web, considerando aspectos como navegación, seguridad, cambios continuos y soporte MDE.
Metodologías (tradicionales, ágiles, híbridas)	Las metodologías tradicionales se enfocan en planificación y procesos definidos; las ágiles priorizan adaptación al cambio; y las híbridas combinan ambos enfoques según el contexto y la experiencia del equipo.
Modelos específicos para web	Incluyen propuestas como OOH, RMM, WebML, SOHDM, OOHDM, UWE y Hera; particularmente, UWE amplía los casos de uso mediante escenarios y glosarios para especificar requisitos.

Continúa en la siguiente página

Tabla XXI (continuación). Conceptos adicionales: síntesis operativa

Concepto	Enfoque / aporte
APIs Web / Descubrimiento automatizado	Permiten recomendar APIs compatibles mediante técnicas de búsqueda por palabras clave y grafos de correlación, como WAR.
Automatización de procesos (RPA)	Facilita la ejecución de tareas repetitivas, reduciendo tiempos operativos, errores y mejorando la experiencia del usuario.
Chatbots con IA	Ofrecen atención continua para consultas frecuentes, orientación y soporte en procesos transaccionales.
Automatización de redes	Permite automatizar configuraciones, pruebas y monitoreo de infraestructura, disminuyendo costos operativos y errores manuales.

Elaboración propia. Fuentes: [37], [38], [39]

2.6.7 Tecnologías de la Nube (Cloud Computing)

El cómputo en la nube ofrece acceso bajo demanda a recursos informáticos con escalabilidad elástica y un modelo de costos de pago por uso. Permite externalizar infraestructura, reducir inversiones de capital y acelerar la entrega de soluciones. Los modelos de servicio más comunes incluyen SaaS, PaaS e IaaS, junto con despliegues híbridos e integraciones con servicios de IA para personalización y analítica [36], [39].

La Tabla XXII sintetiza los modelos y capacidades principales en *cloud computing*.

Tabla XXII: Modelos y capacidades en *cloud computing*

Modelo	**Descripción operativa**	**Ejemplos / casos de uso**
SaaS	Entrega completa aplicación como servicio mientras el proveedor administra la infraestructura y la pila tecnológica.	LMSs, analítica educativa y herramientas de colaboración en línea.
PaaS	Una plataforma administrada para desarrollar, probar y desplegar aplicaciones sin gestionar directamente el sistema operativo u otros detalles del entorno de ejecución.	Backends web, APIs y flujos CI/CD.
IaaS	Recursos escalables bajo demanda en niveles de cómputo, redes y almacenamiento con control sobre el sistema operativo.	Escalado de servicios web y bases de datos administradas por el usuario.
Nube híbrida	Una combinación de entornos privados y públicos de nube diseñada para permitir que las organizaciones balanceen seguridad/conformidad y escalabilidad.	Usar una combinación de ambientes privados y públicos en la nube para aumentar la seguridad mientras se mantienen capacidades de escalado.
Servicios de IA en la nube	Brindan capacidades de análisis de datos y aprendizaje automático como servicios accesibles bajo demanda.	Sistemas de recomendación, segmentación y chatbots inteligentes.

Elaboración propia. Fuentes: [36], [39]

2.7 Computación Serverless

2.7.1 Definición de Computación Serverless

La computación *serverless* es un paradigma de nube que abstrae la provisión y operación de servidores. El proveedor gestiona el aprovisionamiento, el escalado, la alta disponibilidad y el mantenimiento, mientras que el código del usuario se ejecuta bajo demanda, típicamente en respuesta a eventos (p. ej., solicitudes HTTP, cambios en datos o mensajes en colas). La facturación se basa en la duración de la ejecución y en los recursos efectivamente consumidos por invocación [15].

2.7.2 Características de la Computación Serverless

Este modelo se distingue por su orientación a eventos, escalado automático, reducción de tareas operativas y un esquema de costos variable que favorece la agilidad del desarrollo y la eficiencia operativa [15].

La Tabla XXIII resume las características clave del modelo *serverless*.

Tabla XXIII: Características clave del modelo *serverless*

Característica	**Descripción operativa y efectos**
Gestión abstracta del servidor	El proveedor se encarga de administrar la infraestructura, el sistema operativo, los parches y la capacidad, permitiendo que el equipo de desarrollo se concentre en la lógica de negocio.

Continúa en la siguiente página

Tabla XXIII (continuación). Características clave del modelo *serverless*

Característica	**Descripción operativa y efectos**
Escalado automático	La concurrencia se ajusta automáticamente según la cantidad de eventos o solicitudes, respondiendo de manera elástica ante variaciones en la demanda.
Modelo de pago por uso	El costo se calcula según el tiempo de ejecución y los recursos consumidos por invocación, evitando gastos asociados a infraestructura inactiva.
Ejecución orientada a eventos	Permite activar funciones desde colas, almacenamiento, solicitudes HTTP, tareas programadas u otros servicios, favoreciendo arquitecturas *event-driven*.
Alta disponibilidad integrada	La redundancia y los mecanismos de recuperación son administrados por el proveedor para reducir interrupciones del servicio.
Agilidad y *time-to-market*	Reduce la carga operativa y acorta los ciclos de desarrollo, facilitando iteraciones rápidas y despliegues continuos.

Elaboración propia. Fuente: [15]

2.8 Tecnologías Serverless Disponibles

Los principales proveedores que ofrecen capacidades *serverless* integradas con servicios de datos, mensajería y orquestación son Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). Sus catálogos permiten construir aplicaciones orientadas a eventos con almacenamiento y cómputo elástico, adecuadas para cargas intermitentes y picos propios del

ámbito educativo [13].

2.8.1 Amazon Web Services (AWS)

AWS destaca por la madurez de su oferta FaaS (AWS Lambda) y por un ecosistema amplio de integración de eventos, bases NoSQL y colas/temas para desacoplar componentes, lo que facilita el escalado bajo demanda y el pago por uso [5], [13], [15].

En la Tabla XXIV se presentan los servicios de AWS más relevantes para una arquitectura *serverless* orientada a entornos educativos.

Tabla XXIV: Servicios AWS relevantes para una arquitectura *serverless* educativa

Servicio	**Función en la arquitectura / observaciones**
AWS Lambda	Permite ejecutar lógica basada en eventos, como corrección de cuestionarios o cálculo de puntajes, con escalabilidad automática y costos ajustados al tiempo de ejecución y recursos utilizados.
Amazon API Gateway	Facilita la exposición de endpoints HTTP, REST o WebSocket para aplicaciones web o móviles, integrándose de forma directa con AWS Lambda y mecanismos de autorización.
Amazon DynamoDB	Base de Datos NoSQL que proporciona baja latencia en perfiles, progreso académico y tabla de clasificación; y cuenta con autoescalado y arquitecturas orientadas a eventos.
Amazon S3	Servicio de almacenamiento de contenido estático (imágenes), archivos y evidencias, junto a la capacidad de activar eventos de ejecución automática.

Continúa en la siguiente página

Tabla XXIV (continuación). Servicios AWS relevantes para una arquitectura *serverless* educativa

Servicio	Función en la arquitectura / observaciones
Amazon EventBridge	Manejar y ruteo de eventos con base a las reglas programadas para hacer un fácil desacoplamiento entre los modulos y servicios.
Amazon SQS	Servicio de colas orientado a tareas asíncronas y control de picos de procesamiento, incluyendo soporte para *dead-letter queues*.
Amazon SNS	Sistema de publicación y suscripción utilizado para distribuir notificaciones y mensajes hacia múltiples servicios o destinatarios.
Amazon Kinesis Data Streams	Habilita la capacidad de procesar/analizar los datos en tiempo real, especialmente útil para las métricas de uso y la analítica continuada.
AWS IoT Core	Simplifica la toma de datos vía MQTT y su integración en servicios tales como Lambda, DynamoDB cuando se requiere telemetría.
Amazon Cognito	Administra autenticaciones, autorizaciones e identidades de los usuarios, tales como estudiantes y profesores, en la plataforma.

Elaboración propia. Fuentes: [5], [7], [13], [15]

2.8.2 Microsoft Azure

Azure ofrece Azure Functions como FaaS e integra mensajería, datos y gobierno bajo un mismo directorio de identidades. Su propuesta combina orquestación basada en eventos con

opciones PaaS para analítica y almacenamiento, adecuadas para aplicaciones educativas con demanda variable [13], [15].

Las SPA son consideradas “clientes públicos”, ya que no pueden almacenar de forma segura una clave secreta de cliente (*client secret*). Esta característica las expone a riesgos como la interceptación del código de autorización durante el proceso de autenticación.

Con el objetivo de reducir esta vulnerabilidad, el IETF incorporó PKCE (*Proof Key for Code Exchange*) en el RFC 7636 [40]. Este mecanismo añade una capa adicional de verificación mediante dos elementos: el verificador de código (*code verifier*) y el desafío de código (*code challenge*). El cliente genera ambos valores, envía el desafío al iniciar el flujo de autenticación y conserva el verificador de forma local. Posteriormente, cuando el servidor emite el código de autorización, solo permite intercambiarlo por un token si se presenta el verificador correcto. De esta manera, incluso si un atacante intercepta el código de autorización, no podrá utilizarlo sin disponer del valor original [41].

2.8.3 Gestión de Identidades y Seguridad en la Nube (OAuth 2.0 y PKCE)

La gestión de identidades y accesos es esencial en arquitecturas serverless e implementaciones SPA ya que la lógica corre en el browser del usuario. El cliente es un entorno nada de confianza por lo que aumenta la exposición a riesgo. Para solucionar este problema, se utiliza OAuth 2.0, un estándar para permitir el acceso de una aplicación a los recursos protegidos sin el uso de la credencial del usuario. No son las contraseñas sino tokens (como los JWT) los utilizados aquí: credenciales más temporales y con mayores limitaciones [42]. Esto minimiza el peligro de fuga de informaciones sensibles.

El problema es que las SPA son “clientes públicos” (es decir, no podemos de forma segura almacenar de manera persistente un client secret). Esto las deja expuestas a ataques tales como el robo de código de autorización.

Para mitigar esta debilidad, el IETF introdujo Proof key para code exchange (PKCE) del

RFC 7636 [40]. Este mecanismo agrega a la corriente de autenticacion otro nivel de verificaciones con 2 atributos : El 'code verifier' y el 'code challenge'. El cliente crea ambos valores, envía el desafío al comienzo de la transacción, retiene el verificador. Cuando sirve el servidor la autorización del código, sólo te permite intercambiarlo por un token si le presentas al code verifier correctamente. De este modo, incluso si un atacante intercepta el código, no lo utilizará sin dicho valor [41].

Por esta razón, el uso de PKCE es actualmente obligatorio en aplicaciones basadas en navegador según las mejores prácticas del IETF [41]. Además, servicios gestionados como Amazon Cognito permiten implementar este flujo de forma segura y con menor complejidad operativa [43]. En conjunto, OAuth 2.0 define el modelo de autorización y PKCE lo refuerza en entornos donde no es posible proteger secretos, como las SPA.

2.9 Comparativa de Plataformas Serverless

La comparación entre proveedores se enfoca en el desempeño de la mensajería por eventos, los costos asociados a servicios gestionados (p. ej., IoT) y métricas de infraestructura subyacente (CPU, memoria, I/O). A continuación se sintetizan resultados experimentales y hallazgos reportados en la literatura [13], [14].

2.9.1 Rendimiento de servicios IoT (broker MQTT)

Se consideran escenarios punto a punto, *fan-in* y *fan-out*, evaluando métricas como latencia promedio, pérdida de mensajes y comportamiento bajo diferentes niveles de carga. La incorporación de servicios administrados de mensajería permite reducir pérdidas y mantener una entrega más estable de los mensajes.

En la Tabla XXV se resume el rendimiento observado por proveedor en escenarios típicos.

Tabla XXV: Rendimiento MQTT gestionado por proveedor en escenarios típicos

Escenario	**AWS**	**Azure**	**GCP**	**Nota**
Punto a punto	~ 27 ms $\pm$ 0.3 ms (100 clientes); mejor rendimiento entre 150–750 mps.	~ 180 ms $\pm$ 20 ms.	Mayor velocidad fuera del rango 150–750 mps.	El comportamiento depende de la tasa y concurrencia.
Fan-in (múltiples publicadores, 1 suscriptor)	En conexión directa se observan pérdidas desde 400 mps (hasta $\sim 42\,\%$ a 1000 mps). Con SNS/SQS se alcanza 100 % de entrega.	Con Message Bus se mantiene 100 % de entrega y tiempos estables.	Con Pub/Sub se logra 100 % de entrega, aunque el tiempo aumenta después de ~ 3000 mps.	Los buses de mensajería reducen las pérdidas.
Fan-out (1 publicador, múltiples suscriptores)	Menor tiempo de respuesta cuando el número de suscriptores es menor a 15; por encima de ese valor, GCP obtiene mejores resultados.	El reenvío presenta tiempos aproximadamente $15\times$ más altos.	Mejor desempeño cuando el número de suscriptores supera los 15.	Escenario sensible a la cantidad de suscriptores.

Elaboración propia. Fuente: [13]

2.9.2 Costos de servicios IoT

Los modelos de precios difieren según el componente facturable (mensajes, tiempo de conectividad, unidades aprovisionadas o volumen agregado). La conveniencia depende del perfil de tráfico (tamaño de mensaje, ráfagas, número de dispositivos) y del uso de servicios acoplados (reglas, colas, buses). En la Tabla XXVI se resume una comparación de alto nivel reportada en la literatura [13].

Tabla XXVI: Resumen comparativo de costos en IoT gestionado

Dimensión	**AWS IoT Core**	**Azure IoT Hub**	**Google IoT Core**
Unidad de cobro	La facturación considera tiempo de conectividad por minuto, mensajes enviados en bloques de 5 kB y consumo de recursos como Registry, Shadow y Rules.	El cobro está estructurado en las siguientes Unidades Mensuales y Niveles del Servicio (B/S). Cada B/S tendrá Límites Diarios.	La facturación es base a un volumen mesero por dispositivo (>250MB). El tamaño del mensaje mínimo es 1024 bytes.
CRUD de dispositivos	El precio varía con base en la cantidad de operaciones realizadas.	Las operaciones están incluidas dependiendo del nivel o de la unidad contratada.	No tiene un cargo definido ad hoc para tales operaciones.
Mensajería / reenvío	Costos basados en la cantidad de mensajes y uso del motor de reglas; se integra a servicios como SNS y SQS.	La mensajería y el routing se encuentran incluidos dentro de la cuota de la unidad contratada.	La integración a través de pub/sub se centra en costo al volumen que es procesado y al uso en su servicio de mensajería.

Continúa en la siguiente página

Tabla XXVI (continuación). Resumen comparativo de costos en IoT gestionado

Dimensión	AWS IoT Core	Azure IoT Hub	Google IoT Core
Adecuación por perfil de tráfico	Resulta eficiente para volúmenes pequeños o moderados de mensajes, aunque requiere controlar conexiones permanentes.	Adecuado para un costo más predecible en escenarios establecidos, sujeto a picos sobre la límite diario.	Tiene beneficios en escenarios de alto volumen debido a un tamaño mínimo fijo y el aprovechamiento de Pub/Sub.
Sensibilidad a ráfagas	Costos que crecen con los minutos en-línea y también por cada cantidad extra de mensajes enviados.	Límites diarios por unidad puede que requieran incrementos temporales de capacidades (escale-up).	El costo es distribuido sobre el uso de mes.

Elaboración propia. Fuente: [13]

2.9.3 Rendimiento general de Máquinas Virtuales (IaaS)

Comparativas recientes (Ubuntu 20.04 y Phoronix Test Suite) reportan una ventaja consistente de AWS frente a Azure y GCP en CPU, memoria e I/O de disco, con casos puntuales donde GCP lidera en criptografía y lazo de red, y Azure destaca en subpruebas específicas de memoria [14]. En la Tabla XXVII se sintetizan los resultados por dimensión.

Tabla XXVII: Desempeño relativo de VM por proveedor y dimensión

Dimensión de prueba	**AWS**	**Azure**	**GCP**
Puntaje global	87 %	77 %	77 %
CPU (7zip, gzip, kernel, Himeno)	Lidera	Rezaga	Rezaga
Memoria (ramspeed, subpruebas)	Promedio ∼94 %	Lidera en 2 subpruebas	Rezaga
Disco I/O (blogbench, fio)	Superior	Rezaga	Rezaga
Sistema/Cripto (sysbench/cryptsetup)	Alto (no líder)	Alto (no líder)	99 % (líder)
Red *loopback*	Alto (no líder)	Alto (no líder)	100 % (líder)

Elaboración propia. Fuente: [14]

2.9.4 Selección de Amazon Web Services (AWS)

Se adopta AWS como plataforma principal por su desempeño superior en cómputo, memoria e I/O de disco, su escalabilidad bajo demanda, un portafolio amplio y maduro de servicios, y prácticas de seguridad consolidadas. Aunque el costo depende del patrón de uso, la relación rendimiento–confiabilidad–ecosistema favorece a AWS para una solución educativa escalable en picos y eficiente en operación [14]. La Tabla XXVIII resume los criterios y su sustento.

Tabla XXVIII: Criterios de selección de AWS y sustento

Criterio	**Detalle**
Alto rendimiento	Tiene ventajas en procesamientos, memorias y operaciones de lectura/escritura de discos en comparaciones estandarizadas, mantiene siempre los resultados consistentes ante otras alternativas.
Escalabilidad	Puedes escalar tus servicios a escala usando servicios como Aprovisonamiento Elástico,Lambda y DynamoDB, para que ajusten los recursos automaticamente.
Confiabilidad	Proporciona una operación estable y madura bajo diferentes cargas. Mecanismos de alta disponibilidad integrados en el diseño de la infraestructura.
Ecosistema	Cuenta con una amplia gama integrada de servicios que incluyen datos, IA, servicio de mensajería, análisis, lo que facilita el desarrollo y despliegue de soluciones.
Seguridad	Incluye controles de protección de datos avanzados y amplias prácticas adoptadas por la industria para su seguridad.

Elaboración propia. Fuente: [14]

2.10 Evaluación de Usabilidad

2.10.1 Importancia de la Usabilidad

La usabilidad afecta de forma directa a la experiencia de usuario y afecta el rendimiento en las tareas, esta situación evita la realización de las tareas de forma eficiente. En entornos cloud/serverless, su importancia se refleja en la reducción de la carga operativa para los equipos de desarrollo al delegar la gestión de infraestructura, en la disponibilidad de guías prácticas orientadas a la implementación y en el apoyo a la toma de decisiones mediante comparaciones entre plataformas y servicios [15]. La Tabla XXIX sintetiza estos aspectos.

Tabla XXIX: Importancia de la usabilidad en soluciones *cloud*/*serverless*

Aspecto	**Descripción**
Simplificación operativa	Externaliza infraestructura y operaciones (aprovisionamiento, parches, escalado), permitiendo centrarse en la lógica de negocio y reducir la carga operativa del equipo.
Guía práctica	Disponibilidad de directrices detalladas y aplicables para adoptar tecnologías de nube y *serverless* sin sesgos comerciales, acelerando la curva de aprendizaje.
Decisiones informadas	Comparativas de plataformas y *servicios gestionados* ayudan a seleccionar alternativas según el caso de uso, requisitos de rendimiento y costos.

Elaboración propia. Fuente: [15]

2.11 Fundamentos Metodológicos y de Diseño

Para el desarrollo de aplicaciones web modernas que integran estrategias de gamificación y se despliegan sobre arquitecturas en la nube, la literatura académica sugiere la adopción de un enfoque dual. Por un lado, para la gestión del ciclo de vida del desarrollo técnico, se emplean marcos de trabajo ágiles como Scrum [44]. Por otro lado, para el diseño conceptual de la experiencia de usuario, se utiliza el marco Mecánicas-Dinámicas-Estéticas (MDA) [45].

2.11.1 Metodologías Ágiles de Desarrollo de Software

Las metodologías ágiles representan un cambio de paradigma frente a los enfoques tradicionales de desarrollo de software, como el modelo en cascada, que se caracterizan por ser predictivos y secuenciales. En contraste, los métodos ágiles, como Scrum, eXtreme Programming (XP) y DevOps [46], siguen un enfoque colaborativo, iterativo y orientado a las personas [44], reflejando los principios del Manifiesto Ágil. Este manifiesto prioriza a los individuos y sus interacciones sobre los procesos y herramientas, el software funcional sobre la documentación exhaustiva, la colaboración con el cliente sobre la negociación de contratos y la respuesta ante el cambio sobre el seguimiento de un plan estricto.

Según la literatura existente, hay 3 razones principales para su uso en proyectos de software:

1. **Adaptabilidad a requisitos cambiantes:** Las metodologías ágiles acogen el cambio de requisitos, aunque sea tardío en el desarrollo. El proceso es impulsado por retroalimentaciones de los usuarios.

2. **Entrega incremental de valor:** No pueden esperar hasta que tengamos el producto final completo. Tenemos que concentrarnos en la entrega de incrementos del software funcional en intervalos cortos [44].

3. **Enfoque en la colaboración:** Promueven la interacción entre los desarrolladores y los stakeholders frecuente y directa [44].

2.12 El Marco de Trabajo Scrum

Entre los distintos métodos ágiles, Scrum es el más popular y adoptado en la industria [44], [46], con un 81 % de las organizaciones reportando su uso [44]. Su popularidad se atribuye a su simplicidad y a su capacidad para mejorar la productividad de los equipos frente a arquitecturas complejas [47].

Scrum no es una metodología prescriptiva acerca de cómo llevar a cabo cada paso en la programación. Es un marco de trabajo empírico con estructuras de rol, evento y artefacto. Esto sirve para manejar el desarrollo del producto iterativamente y adaptablemente. Su meta es crear valor a partir de las ideas por medio de Sprints.

2.12.1 Roles Teóricos del Equipo

El marco Scrum se compone de tres roles principales que rigen el ciclo de vida del producto [44]:

- ***Product Owner*:** Define la visión del producto, maximiza el valor del trabajo y prioriza las funcionalidades del *Backlog*.
- ***Developers*:** Diseñan la arquitectura, programan, prueban y despliegan los incrementos de la aplicación en cada ciclo.
- ***Scrum Master*:** Facilita la correcta aplicación del marco de trabajo, asegura la transparencia y remueve impedimentos técnicos.

2.12.2 Eventos y Artefactos de Scrum

Scrum prescribe cinco eventos recurrentes que fomentan la inspección y adaptación:

- ***Sprint*:** Un sprint tiene marco temporal estricto para producir un incremento utilizable y probado.
- ***Sprint Planning*:** Definimos objetivo del sprint y el plan de trabajo.
- ***Daily Scrum*:** Sincronización diaria para evaluar progreso.
- ***Sprint Review*:** Inspección del incremento finalizado.
- ***Sprint Retrospective*:** Se realiza para identificar el mejoramiento existente en el proceso de ingeniería.

Estos eventos están soportados por tres artefactos que dan certeza a la transparencia del avance: un Product Backlog (La lista maestra de requerimientos), un Sprint Backlog (tareas para la iteración actual) y un incremento (Resultado potencialmente enviable).

2.12.3 El Product Backlog y las Historias de Usuario

El *Product Backlog* es un artefacto dinámico que contiene una lista priorizada de todos los requisitos del sistema [44]. Para la especificación de estos requisitos en entornos ágiles, se emplean las Historias de Usuario (HU), las cuales son descripciones breves y sencillas de una funcionalidad contadas desde la perspectiva del usuario final [46].

Alami [47] dice que una historia de usuario se tiene que escribir en forma estándar: Como [rol], quiero [función] por [beneficios].” Y, por último, pero no menos importante, una HU requiere acompañarse con Criterios de Aceptación para ser considerada completa y verificable. Los Criterios de Aceptación determinan las fronteras de la historia e indican como saber si la funcionalidad es implementada adecuadamente (Técnicamente y en el contexto del Negocio).

2.12.4 Estimación de Esfuerzo: Story Points

A diferencia de los métodos tradicionales que estiman en horas, Scrum utiliza los *Story Points* (Puntos de Historia). Estos representan una medida relativa del esfuerzo, complejidad y riesgo necesario para completar una Historia de Usuario [44]. Esta unidad de medida permite al equipo calcular su velocidad y predecir tiempos de entrega de manera más precisa en entornos de alta incertidumbre técnica.

2.13 Gamificación y el Marco MDA

Mientras Scrum organiza la construcción de la arquitectura de software, el marco MDA (Mecánicas, Dinámicas, Estéticas) define el diseño de la experiencia de usuario gamificada [45], [48], [49]. Este marco descompone la experiencia en tres niveles de abstracción y establece que el diseño debe realizarse a la inversa: primero se definen las emociones, luego el comportamiento y finalmente las reglas de código [45].

1. **Mecánicas (*Mechanics*):** Reglas duras, Componentes y Algoritmos Base que subyacen a todo el sistema. (ej. cálculos de puntuación, restricciones de tiempo).
2. **Dinámicas (*Dynamics*):** Representaciones sobre el comportamiento de un sistema en tiempo de ejecución inducido por las mecánicas (p.e.: acción, movimiento). progresión de dificultad, sentido de competencia).
3. **Estéticas (*Aesthetics*):** Respuestas emocionales y afectivas que la experiencia evoca en el usuario final (Ejemplo: atractivo, sensación de logro, orgullo, desafío intelectual).

2.14 Ingeniería de Software y Modelado con UML

Para la formalización de la plataforma educativa PreServerless, se adopta el Lenguaje de Modelado Unificado (UML), definido por Jiménez de Parga [50] como el lenguaje estándar de

la industria para visualizar, especificar y documentar los modelos que conforman un sistema de software. Según el autor, UML no es simplemente un conjunto de diagramas, sino un lenguaje para la elaboración de planos de software que permite abstraer la complejidad de arquitecturas modernas mediante una notación gráfica común.

2.14.1 Categorización de Diagramas UML

La metodología propuesta por Jiménez de Parga divide los diagramas en dos grandes grupos: estructurales (estáticos) y de comportamiento (dinámicos) [50]. Para este proyecto, se han seleccionado diagramas específicos de ambas categorías que permiten cubrir desde la interacción del estudiante hasta la infraestructura física en la nube de AWS.

2.14.2 Modelado Funcional: Diagramas de Casos de Uso

El diagrama de casos de uso representa la funcionalidad del sistema desde una perspectiva externa. Jiménez de Parga destaca que estos diagramas permiten identificar a los actores (entidades externas que interactúan con el sistema) y los casos de uso (unidades de funcionalidad) [50]. En la presente investigación, este modelo define las fronteras del sistema y las interacciones del Estudiante y Administrador con los servicios de autenticación y evaluación.

2.14.3 Modelado Dinámico: Diagramas de Secuencia y Actividad

Estos modelos atrapan la lógica de procedimiento y tiempo del software:

- **Diagrama de Secuencia:** Describe la interacción de los objetos organizados en un orden temporal [50]. Es crucial para este trabajo con respecto al modelado del intercambio de mensajes. Comunicación asíncrona entre el navegador web, Amazon Cognito y las Funciones de AWS Lambda.

- **Diagrama de Actividad:** Se utiliza para modelar los aspectos dinámicos del sistema mediante flujos de control paso a paso [50]. Permite visualizar procesos paralelos y bifurcaciones, siendo esencial para formalizar el motor de evaluación concurrente (control de tiempo y renderizado de reactivos).

2.14.4 Modelado Estructural: Diagramas de Clases y Despliegue

Formalizan la organización de los datos y la topología física:

- **Diagrama de Clases (Perspectiva Lógica NoSQL):** Define la estructura de las entidades. Aunque tradicionalmente orientado a objetos, en este proyecto se adapta bajo el enfoque de Jiménez de Parga para representar el esquema lógico de los documentos almacenados en Amazon DynamoDB, detallando atributos polimórficos y claves de acceso [50].
- **Diagrama de Despliegue:** Ilustra la arquitectura física del hardware y los artefactos de software que residen en ellos [50]. Define la topología de la infraestructura gestionada, especificando los nodos de ejecución cliente (browser) y servidor (nube AWS).

A continuación, la Tabla XXX resume el propósito técnico de cada diagrama empleado, tomando como base los planteamientos de Jiménez de Parga.

Tabla XXX: Resumen de diagramas UML aplicados según Jiménez de Parga

Diagrama	**Aporte técnico al proyecto**
Casos de Uso	Permiten identificar los requerimientos funcionales del sistema y delimitar las interacciones de los roles Estudiante y Administrador.

Continúa en la siguiente página

Tabla XXX (continuación). Resumen de diagramas UML aplicados según Jiménez de Parga

Diagrama	**Aporte técnico al proyecto**
Actividad	Describe el algoritmo de evaluación y la lógica de control asociada al temporizador dentro del flujo de ejecución.
Secuencia	Representa el intercambio de mensajes asíncronos y el proceso de validación de tokens JWT entre componentes.
Clases / Lógico	Define las estructuras de datos persistidas en tablas NoSQL, incluyendo usuarios, preguntas y resultados.
Despliegue	Muestra la topología física de los servicios *serverless* y la comunicación segura mediante protocolos HTTPS/TLS.

CAPÍTULO III

MATERIALES Y MÉTODOS

Materiales

En este capítulo se describen de manera detallada los recursos tecnológicos y servicios de nube utilizados en el desarrollo de la aplicación web educativa gamificada. La selección de los materiales buscó garantizar escalabilidad, disponibilidad, bajos costos de operación y facilidad de mantenimiento. La solución se implementó bajo el paradigma *serverless*, aprovechando servicios totalmente administrados para eliminar la gestión de servidores y concentrar los esfuerzos en la lógica de negocio y la experiencia del usuario.

3.1 Patrones de Diseño Arquitectónico

Bajo las premisas del Entorno Cloud Computing, el diseño de la plataforma está sustentado sobre dos patrones arquitectónicos principales: La arquitectura orientada a eventos o EDA por sus siglas en inglés; y el patrón de microservicios Serverless.

El patrón EDA por un lado deja que el sistema responda asíncronamente a la interacción del usuario. A diferencia de mantenerse con procesos ejecutándose continuamente, los componentes de cómputo (AWS Lambda) son invocados (triggereados) sólo cuando aparecen ciertos eventos, como es una request HTTP entrante canalizada por medio de Amazon API Gateway o un token en Amazon Cognito se ha validado. Con este enfoque se logra un buen desacoplamiento entre el frontend SPA y la lógica de negocio, así como la optimización del recurso operacional bajo Pago por Uso.

De la otra parte se llevó a cabo una aproximación granular de Microservicios. A diferencia

de una arquitectura monolítica tradicional: La lógica del backend fue dividida en seis funciones autónomas e independientes (ej. obtenerPreguntas, guardarResultados, getMetrics). Cada microservicio tiene una única responsabilidad, opera aislado, tiene su propio ciclo de vida de despliegue usando node js 20.x, se comunica a la base de datos nosql(Amazon Dynamo DB) cumpliendo los principios mínimos de privilegios vía roles IAM. Esta segmentación hace fácil mantener este software, disminuye los efectos del fault isolation y deja cada módulo escalar horizontalmente de forma autónoma cuando se incrementa la concurrencia de aspirantes.

3.2 Arquitectura General de la Solución

Esta aplicación fue diseñada con arquitectura en capas para promover la separación de responsabilidades y un fácil mantenimiento. Capas: El sitio web estático (HTML, CSS, JS) que se despliega por medio de AWS S3 es la capa de presentación. La capa de identidad y seguridad está manejada con Amazon Cognito. La comunicación fue implementada mediante Amazon Api Gateway - exponiendo los recursos REST de la interacción front/backend. La capa de negocio se compone de funciones de AWS Lambda, que están escritas en node.js 20.x y ejecutan la lógica de la aplicación. La capa de persistencia finalmente emplea Amazon DynamoDB y tiene tres tablas especializadas.

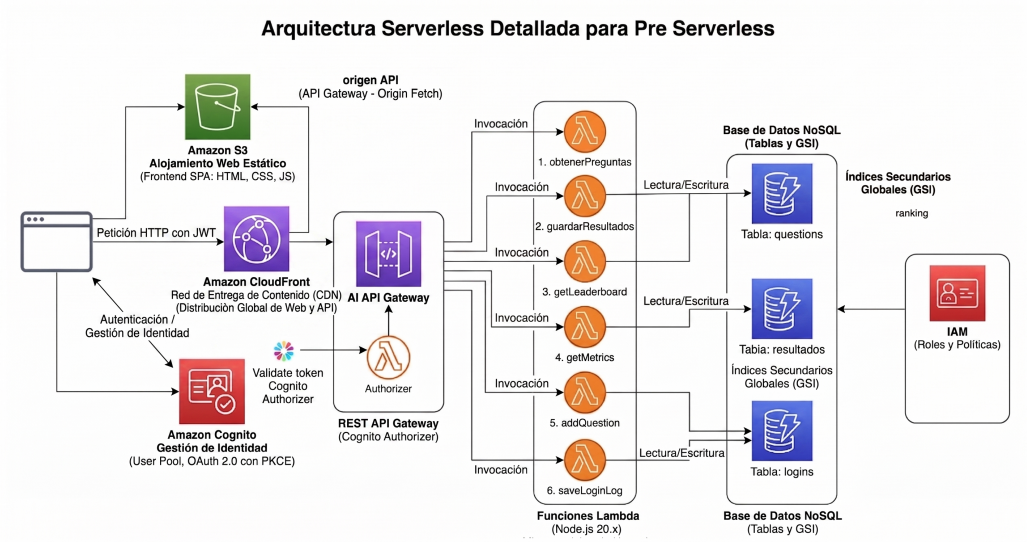
El flujo general de interacciones continúa en esta secuencia:

1. El Usuario accede a la URL Pública de la página Web alojada en Amazon S3.
2. Autenticación del Usuario: El usuario se autentifica con Amazon Cognito usando el flujo OAuth 2.0 con PKCE y obtiene los tokens JWT.
3. Se hacen peticiones HTTP desde el front-end hacia los ’endpoints’ del .ªPI Gateway”, con el ”token.en la cabecera.
4. Autorizador de API gateway autentica a los usuarios y llama a las lambdas correctas.

5. El conjunto de consultas Lambda son escritas o leídas a la tabla de DynamoDB (preguntas y/o resultados).

6. Entonces, la respuesta se devuelve al frontend en formato json para mostrarla en la interfaz.

Figura 1.
Arquitectura *serverless* implementada en AWS para la aplicación web educativa gamificada.



*La **Tabla XXXI** resume los componentes principales de la arquitectura y la función que desempeñan dentro de la solución propuesta.*

Tabla XXXI: Componentes principales de la arquitectura *serverless*

Componente	**Rol en la solución**
Frontend (HTML, CSS, JS)	Interfaz interactiva de usuario basada en SPA, encargada del renderizado dinámico de exámenes, validaciones y consumo de APIs.
Amazon S3	Servicio utilizado para el almacenamiento y publicación del sitio web estático.

Continúa en la siguiente página

Tabla XXXI (continuación). Componentes principales de la arquitectura

Componente	**Rol en la solución**
Amazon Cognito	Gestiona identidades de usuarios mediante *User Pool*, autenticación con OAuth 2.0 y PKCE, además de control de acceso basado en roles.
Amazon API Gateway	Permite exponer recursos REST, administrar rutas, habilitar CORS y proteger endpoints mediante *Cognito Authorizer*.
AWS Lambda (Node.js 20.x)	Implementa seis microservicios de negocio orientados a obtención de preguntas, almacenamiento de resultados, rankings, métricas administrativas, carga masiva en JSON y auditoría.
Amazon DynamoDB	Base de datos NoSQL compuesta por las tablas `questions`, `resultados` y `logins`, optimizadas mediante Índices Secundarios Globales (GSI).
IAM (roles y políticas)	Gestiona los permisos para ejecutar código y acceder a recursos de acuerdo con el menos privilegio.

Formalización UML: Diagrama de Despliegue Para complementar el esquema de servicios cloud (Figura 1) a estándares de la ingeniería de software se preparó el Diagrama de despliegue UML (Figura 2). Este modelo describe la topología física del sistema, definiendo abiertamente los nodos de ejecución (dispositivo cliente, Localizaciones frontera de S3, contenedores efímeros de Lambda). Se despliegan artefactos de software dentro y se utiliza una capa de seguridad para asegurar la integración de datos usando un protocolo seguro (HTTPS/TLS) sobre un entorno web gestionado.

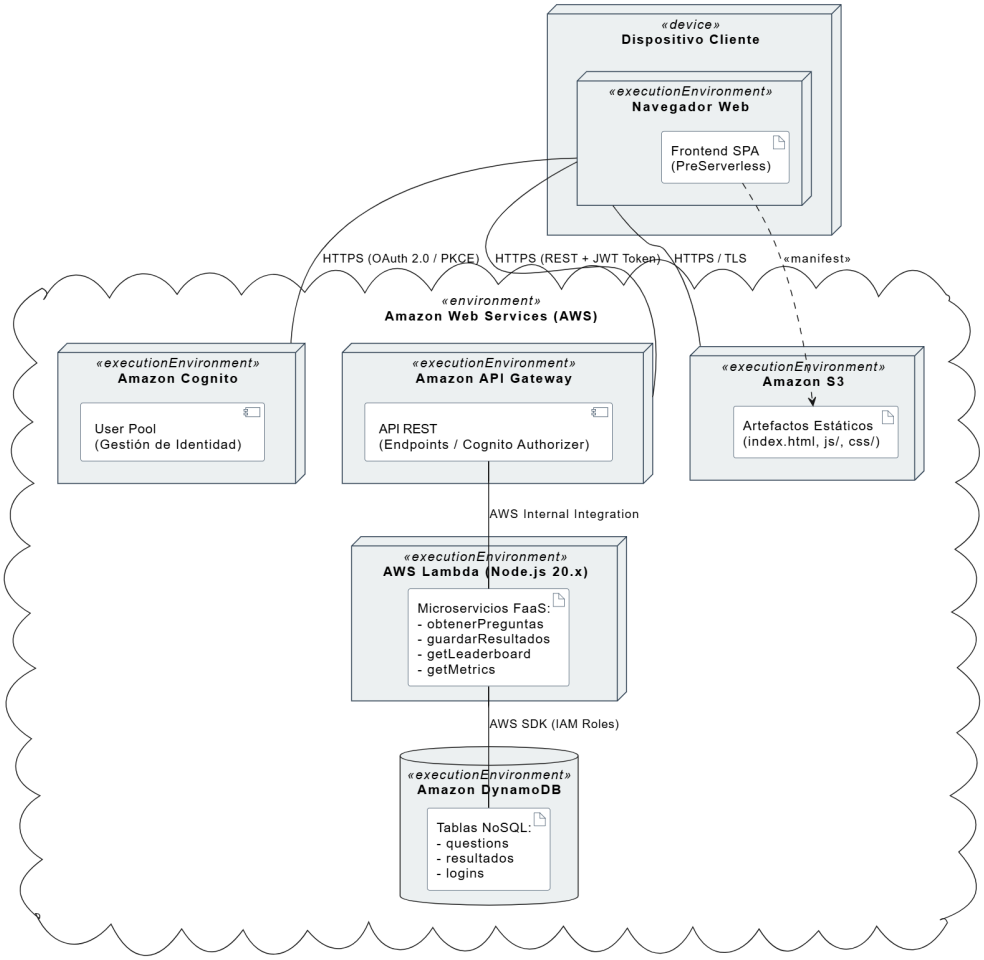


Figura 2.
Diagrama de Despliegue UML de la infraestructura Serverless.

3.3 Amazon API Gateway: Recursos y Seguridad

La comunicación fue realizada a través de una interfaz REST protegida. API Gateway actúa como punto de entrada único, asegurándose que solamente los usuarios autenticados tengan acceso a las funciones de evaluación/administración.

Tabla XXXII: Recursos REST publicados en Amazon API Gateway

Recurso REST	Propósito funcional	Método	Integración Backend
`/questions`	Entrega preguntas aleatorias según categoría y dificultad.	GET	Lambda: `obtenerPreguntas`
`/submit`	Registrar Resultado Del Intento, Puntaje Y Detalle del Examen.	POST	Lambda: `guardarResultados`
`/leaderboard`	Retorna el rankeo de scores ordenados utilizando GSI del DynamoDB.	GET	Lambda: `getLeaderboard`
`/admin/metrics`	Calcula estadísticas de embudo, retención y sus errores frecuentes.	GET	Lambda: `getMetrics`
`/add-questions`	Acepta carga masiva de bancos de ítems vía archivo JSON.	POST	Lambda: `addQuestion`
`/log-login`	Registre los meta datos de logueo para auditoria en la seguridad.	POST	Lambda: `saveLoginLog`

3.4 Entorno de desarrollo y herramientas de soporte

Herramientas modernas fueron usadas para la construcción de la plataforma; herramientas que permiten diseñar un sitio responsivo o ver los datos analíticos en tiempo real.

Tabla XXXIII: Herramientas de desarrollo, librerías y propósito

Herramienta / Librería	Uso en el proyecto
Node.js 20.x	Entorno de ejecución Node.js 20.x para los seis servicios microservicios AWS lambda.
Tailwind CSS	CSS Tailwind CSS Framework CDN integrado, diseño web modernista y responsivo.
Chart.js	Dinamización gráfica de gráficas de dona y barras en el panel del administrador.
jsPDF y AutoTable	Generación y exportación de reportes tabulares de métricas en formato PDF.
AWS CLI	Gestión y provisionamiento de la infraestructura en la nube a través de comandos de consola AWS CLI.

3.5 Estructura de artefactos del frontend

Cómo fueron organizados los archivos en el bucket de S3. La lógica y interfaces de usuario fueron separados con este fin en mente del mantenimiento evolutivo.

Tabla XXXIV: Organización de artefactos del frontend

Ruta o archivo	Contenido o propósito
`/welcome.html`	Página de aterrizaje (Landing Page) y Acceso delegado a Cognito /welcome.html.
`/index.html`	Panel principal del estudiante post-autenticación.
`/seleccion.html`	Elegir la área académica y el nivel de dificultad.

`/prueba.html`	Dinámicamente prueba ítem mcq, complete y emparejamiento usando motor evaluador: /prueba.html.
`/retroalimentacion.html`	Desglose de aciertos, fallos y justificaciones pedagógicas.
`/resultado.html`	Visualización del ranking global de aspirantes.
`/admin.html`	Exclusivo panel administrativo: métricas, gráficas, carga JSON y más.
`/js/`	Autentificación PKCE (auth.js), Servicios APIs (api.js) y controladores.

3.6 Estructuras de datos y persistencia

La solución planteada utiliza Amazon DynamoDB a manera de motor de persistencia NoSQL. La tabla XXXV simplifica el esquema del diseño que está orientado a los documentos del banco de preguntas.

Tabla XXXV: Campos lógicos del ítem de pregunta para carga en JSON

Campo	**Tipo de Dato**	**Descripción Lógica**
`id`	String	Identificador único alfanumérico generado en el *backend*.
`tipo`	String	Formato del ítem: `mcq`, `completacion` o `matching`.
`categoria`	String	Área de conocimiento (*e.g.*, verbal, matemáticas).

`enunciado`	String	Instrucción principal o planteamiento del problema.
`opciones`	Lista(String)	Alternativas de respuesta (Solo para tipo `mcq`).
`correcta`	Number / String	Índice de la opción correcta o palabra clave esperada.
`explicacion`	String	Texto pedagógico mostrado en la retroalimentación.

El esquema de la base de datos NoSQL se diseñó en base a los patrones de acceso de la aplicación. Al utilizar Amazon DynamoDB, los documentos no tienen un esquema rígido, pero requieren la definición estricta de una Clave de Partición (PK) y, opcionalmente, una Clave de Ordenamiento (SK) para garantizar un rendimiento óptimo en milisegundos.

Además del banco de reactivos, el modelo de persistencia se complementa con la tabla de `resultados`, la cual almacena el historial detallado de intentos por usuario para alimentar la lógica de gamificación. Su estructura se detalla en la Tabla XXXVI.

Tabla XXXVI: Estructura lógica de la tabla de resultados (`resultados`)

Campo	**Tipo de Dato**	**Descripción Lógica**
`usersId` (PK)	String	Identificador único del estudiante (sub de Cognito).
`attemptId` (SK)	String	Identificador único del intento (prefijo `att_` + timestamp).

`leaderboardKey`	String	Clave compuesta (`categoria#dificultad`) para indexación en el GSI de ranking.
`detalle_examen`	String	Almacenamiento en formato JSON del mapa de respuestas para retroalimentación asíncrona.
`puntaje`	Number	Resultado numérico final calculado por el motor de evaluación.
`timeSpentSec`	Number	Tiempo total invertido en la resolución del examen (en segundos).
`startedAt`	String	Marca de tiempo utilizada como clave de ordenamiento en el índice secundario.

Finalmente, para cumplir con los requerimientos de auditoría y trazabilidad de seguridad (CU-02), se implementó la tabla `logins`, encargada de registrar los metadatos de acceso, tal como se especifica en la Tabla XXXVII.

Tabla XXXVII: Estructura lógica de la tabla de auditoría de accesos (`logins`)

Campo	**Tipo de Dato**	**Descripción Lógica**
`userId` (PK)	String	Identificador único del usuario autenticado en la sesión.
`loginTime` (SK)	String	Marca de tiempo de la ejecución del evento de inicio de sesión.
`ipAddress`	String	Dirección IP de origen capturada desde el objeto *request* de API Gateway.

`userAgent`	String	Cadena de texto contiene información del navegador y del sistema operativo del cliente.

3.7 Desempeño, escalabilidad y costos operativos

Se analizaron los parámetros de rendimiento bajo carga y los costos asociados con el modelo de pago por uso de AWS para asegurar que el proyecto sea viable.

Tabla XXXVIII: Consideraciones de rendimiento y escalabilidad

Componente	**Mecanismo de respuesta y escalado**
Alojamiento estático en S3	Estático alojamiento en S3 Entrega con baja latencia de los archivos con escala automática.
AWS Lambda (FaaS)	Escalado horizontal automático para invocar una unidad concurrente.
DynamoDB (On-demand)	Ajuste dinámico del ajuste de la capacidad de lectura/escritura como se demande.

Como arquitectura no requiere del aprovisionamiento de capacidad ociosa, los costos operativos permanecen dentro de un rango marginal; lo cual es la sostenibilidad económica ideal para un centro educativo creciente.

3.8 Riesgos técnicos y mitigaciones

Se identificaron riesgos inherentes a la infraestructura en la nube, estableciendo estrategias de mitigación centradas en la seguridad y la disponibilidad.

Tabla XXXIX: Principales riesgos y acciones de mitigación

Riesgo	**Mitigación**
Errores de CORS en el navegador	Configuración estricta de cabeceras en API Gateway y Lambdas para restringir el acceso solo desde dominios autorizados.
Picos de tráfico simultáneo	Uso de servicios totalmente gestionados con escalado elástico nativo de AWS (Lambda, DynamoDB On-demand).
Pérdida de conectividad en el dispositivo cliente	Implementación de una cola de persistencia local (`LocalStorage`) que almacena los resultados para su reenvío automático ante fallas de red.
Acceso no autorizado a funciones de gestión	Validación obligatoria de la pertenencia al grupo `superuser` en los *claims* del token JWT dentro de la lógica del *backend*.
Inconsistencias en rankings globales	Uso de Índices Secundarios Globales (GSI) con consistencia eventual y validación de integridad de *payloads* antes de la persistencia.
Permisos excesivos en recursos de nube	Aplicación de políticas de IAM granulares bajo el principio de mínimo privilegio para cada ejecución de microservicio.

3.9 Glosario de siglas y términos

Para asegurar la correcta interpretación de los tecnicismos utilizados en este capítulo, se presenta la Tabla XL.

Tabla XL: Glosario de términos técnicos utilizados

Sigla o término	Definición operacional
JWT	JSON Web Token; estándar para la transmisión segura de identidad.
PKCE	Proof Key for Code Exchange; flujo de seguridad para aplicaciones SPA.
GSI	Global Secondary Index; optimización de búsquedas en DynamoDB.
FaaS	Function as a Service; ejecución de código por eventos en la nube.
SPA	Single Page Application; aplicación web de una sola página.

3.10 Cierre del capítulo de materiales

Integra el conjunto de materiales descritos: Frontend Interactivo en S3, Gestión de Identidad (Amazon Cognito), Capa Orquestración (API Gateway) , Lógica Negocio Robusta (Lambda) + Persistencia (DynamoDb). Esta composición proporciona una base segura, mantenible y profesional para la plataforma educativa PreServerless.

Métodos

3.11 Síntesis de la Metodología Integrada

La metodología de este proyecto se articula en dos capas inseparables. En la capa de ejecución técnica, Scrum proporciona el rigor de la ingeniería de software para desarrollar micro-servicios (*FaaS*) de manera segura e iterativa. En la capa de diseño de interacción, el marco MDA garantiza que cada línea de código (Mecánica) tenga un propósito pedagógico directo que fomente la motivación (Estética). Esta sinergia es el pilar para desarrollar *Pre Serverless* como una herramienta técnicamente impecable y educativamente transformadora.

3.11.1 Estrategia de Búsqueda y Trazabilidad Bibliográfica

Para fundamentar el diseño y desarrollo de la aplicación web educativa, se estructuró una revisión de literatura rigurosa. El propósito de esta fase fue garantizar la trazabilidad metodológica entre los objetivos específicos del proyecto y el estado del arte actual en computación en la nube y tecnologías educativas.

3.11.1.1 Cadenas de Búsqueda y Bases de Datos

La revisión bibliográfica no se apoyó en motores de búsqueda genéricos, precisamente para evitar el ruido documental que suele mezclar investigaciones rigurosas con contenido de escasa validez académica. En su lugar, la búsqueda se concentró en repositorios científicos de alto impacto y bases de datos especializadas como IEEE Xplore, Scopus, SpringerLink, además de plataformas multidisciplinares reconocidas como MDPI y Frontiers. La intención fue construir un marco teórico sólido, sustentado en publicaciones con procesos formales de revisión y validación científica.

Para garantizar la pertinencia de los resultados, se diseñaron cadenas de búsqueda booleanas aplicadas sobre metadatos específicos, como título, resumen y palabras clave, permitiendo filtrar estudios relacionados directamente con arquitecturas tecnológicas, computación *serverless*, escalabilidad en la nube y enfoques pedagógicos asociados a la gamificación educativa. Asimismo, la consulta se restringió a publicaciones recientes, priorizando investigaciones alineadas con el estado actual de las tecnologías Cloud y las tendencias contemporáneas en aprendizaje digital. El objetivo no fue acumular referencias de

manera indiscriminada, sino seleccionar trabajos que aportaran sustento conceptual y coherencia técnica al desarrollo del proyecto.

Tabla XLI: Cadenas de búsqueda booleanas aplicadas por área de estudio

Eje Temático	**Cadena de Búsqueda (Search String)**
Serverless Computing y Arquitectura Cloud	(“Serverless computing” OR “FaaS” OR “Cloud native”) AND (“Web architecture” OR “AWS”) AND (“Microservices” OR “Scalability”)
Gamificación y Aplicaciones Educativas	(“Gamification” OR “Game-based learning”) AND (“Education” OR “E-learning” OR “Higher education”) AND “Web application”
Evaluación de Usabilidad	(“System Usability Scale” OR “SUS”) AND (“User experience” OR “Web interface evaluation”)

3.11.1.2 Revistas Científicas y Autores Representativos

Con el propósito de fortalecer la validez y confiabilidad del marco conceptual, la selección bibliográfica se restringió a publicaciones científicas sometidas a procesos de revisión por pares (*peer-review*) y pertenecientes a revistas con indicadores de impacto reconocidos dentro de sus respectivas áreas de conocimiento. Este criterio permitió priorizar investigaciones con mayor rigor metodológico y relevancia académica, evitando fuentes de carácter divulgativo o con escaso respaldo científico. Entre las principales revistas que sustentan esta investigación se destacan:

- **Ingeniería y Computación:** *IEEE Transactions on Software Engineering*, *IEEE Access*, *Journal of Cloud Computing*, *Information and Software Technology*.
- **Educación y Psicología:** *Education Sciences*, *Frontiers in Psychology*, *Computers in Human Behavior*.

- **Documentación Técnica de la Industria:** Documentación oficial y Whitepapers de *Amazon Web Services (AWS)* y el *Internet Engineering Task Force (IETF)*.

Autores como Hassan et al. [4] y Jonas et al. [5] fueron referenciados para consolidar las definiciones de infraestructura *serverless*; mientras que para la sustentación del modelo gamificado se emplearon investigaciones clave como las de Li et al. [1], Krath et al. [26] y Lampropoulos [8].

3.11.1.3 Matriz de Trazabilidad Bibliográfica

La integración de las fuentes consultadas responde directamente a los objetivos específicos planteados para la solución tecnológica. La Tabla **??** mapea la bibliografía utilizada con los entregables del proyecto, demostrando que ninguna tecnología o teoría fue adoptada sin respaldo científico.

Tabla XLII: Trazabilidad: Del Marco Conceptual a la Arquitectura Funcional

Componente del Objetivo	**Sustento Teórico (Marco Conceptual)**	**Implementación Técnica (Arquitectura)**
1. Marco conceptual: Serverless, apps educativas y gamificación funcional.	Fundamentos de computación *serverless* [4], [15] y modelos de gamificación [1], [8].	Definición de servicios AWS y patrones de diseño (FaaS y EDA).
2. Arquitectura funcional: Integración de servicios Cloud (FaaS, BD, almacenamiento).	Patrones de microservicios y escalabilidad bajo demanda [5], [9].	Uso de AWS Lambda (Node.js), API Gateway y S3.
3. Persistencia escalable: Bases de datos administradas.	Modelado de datos NoSQL para alta concurrencia [7].	Implementación en Amazon DynamoDB con GSI para *leaderboards*.

Componente del Objetivo	Sustento Teórico	Implementación Técnica
4. Seguridad y acceso: Modelo integrado de servicios.	Estándares de autenticación delegada OAuth 2.0 y PKCE [40], [42].	Integración de Amazon Cognito con API Gateway (Autorizador JWT).

3.12 Requerimientos del Sistema

La ingeniería de requisitos definió el alcance técnico de la plataforma gamificada. A diferencia de un desarrollo tradicional, los requerimientos se redactaron considerando las capacidades y restricciones del entorno *Serverless* de AWS.

3.12.1 Requerimientos Funcionales (RF)

Los requerimientos funcionales detallan las operaciones que el sistema debe ejecutar de forma obligatoria para satisfacer el diseño pedagógico (MDA).

Tabla XLIII: Matriz de Requerimientos Funcionales

ID	Módulo	Descripción Técnica
RF-01	Seguridad y Acceso	El sistema debe autenticar usuarios mediante Amazon Cognito usando el flujo OAuth 2.0 con PKCE, devolviendo un token JWT válido.
RF-02	Motor de Evaluación	El Backend tiene que proveer de JSON. Este Json debe tener un arreglo randomizado de 10 reactivos acorde al nivel.

RF-03	Motor de Evaluación	El sistema debe procesar la calificación de forma inmediata en el cliente para retroalimentación asíncrona y persistir los resultados de forma segura en AWS Lambda.
RF-04	Ranking Global (GSI)	El sistema debe consultar y ordenar descendentemente los puntajes mediante un Índice Secundario Global para asignar medallas.
RF-05	Analítica Administrativa	El panel debe graficar mediante *Chart.js* las métricas de rendimiento y permitir la descarga del reporte en formato PDF.
RF-06	Carga de Reactivos	El sistema debe permitir la inserción masiva de nuevas preguntas mediante la subida de un archivo en formato `.json`.

3.12.2 Requerimientos No Funcionales (RNF)

Definen las limitaciones métricas y los atributos de calidad para permitir la factibilidad del funcionamiento operacional en la nube.

Tabla XLIV: Matriz de Requerimientos No Funcionales

ID	Atributo	Especificación
RNF-01	Rendimiento	La latencia de respuesta de las funciones Lambda (en caliente) no debe superar los 500 milisegundos para evitar bloqueos en el DOM.

RNF-02	Disponibilidad	La arquitectura debe ser tolerante a fallos (*Fault Tolerant*) delegando el tiempo de actividad al SLA de infraestructura de AWS (99.9 %).
RNF-03	Escalabilidad	Los microservicios deben escalar horizontalmente de forma automática (On-demand) ante picos de concurrencia de aspirantes.
RNF-04	Seguridad	La transmisión de datos emplea protocolo TLS 1.2+. La capa de negocio (Lambda) valida estrictamente el grupo `superuser` en los *claims* del token JWT para restringir el acceso a métricas y gestión de reactivos.

3.13 Modelado Funcional: Especificación de Casos de Uso

En aras de formalizar el comportamiento del sistema y de conseguir trazabilidad entre las historias de usuario (scrum) y la implementación técnica fue que se pasó a caso de uso. Anteriormente a la definición de los casos, fueron definidos los actores del sistema.

3.13.1 Identificación de Actores

- **Estudiante (Actor Principal):** Interacciona con la Plataforma de Frontend: Realización de Simulacros, Sométase usted mismo a Evaluación con Límite de Tiempo y consulte su posición en ranking gamificado..
- **Administrador (Actor de Gestión):** Personal académico o técnico con elevados derechos en su ámbito. JWT token, auditor de errores, responsable para update del banco de reactivo NoSQL.
- **AWS (Actor de Sistema):** Como componente administrado del sistema se incluyen componentes AWS: Amazon Cognito o API Gateway actúan por si solos.

3.14 Casos de Uso

3.14.1 CU-01: Autenticación Segura y Delegada

Para este caso de uso interviene el actor Estudiante y Administrador, los cuales podrán solicitar el acceso al sistema de forma segura. En la figura 3 se observa este caso de uso.

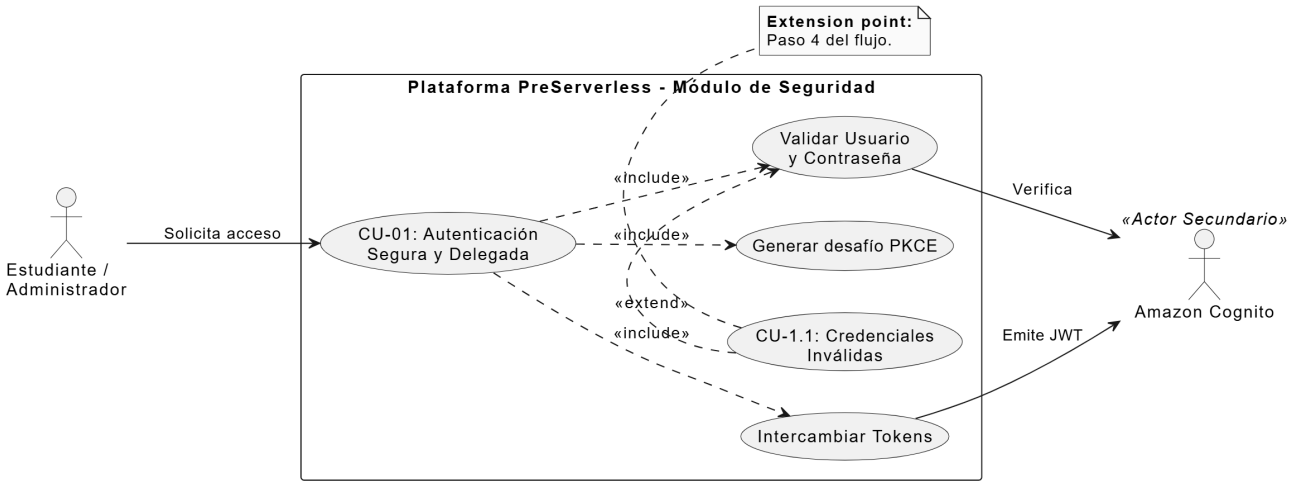


Figura 3.
Caso de uso Autenticación

A continuación, en la tabla XLV se especifican los actores, precondiciones y los pasos del flujo principal necesarios para que los usuarios accedan al sistema de forma segura.

Tabla XLV: Especificación de CU-01: Autenticación Segura y Delegada

Atributo	Descripción
Identificador	1
Breve descripción	El usuario accede al sistema de forma segura mediante credenciales centralizadas utilizando el estándar OAuth 2.0 con PKCE.
Actores primarios	Estudiante, Administrador.
Actores secundarios	Amazon Cognito.

Tabla XLV: Especificación de CU-01 (Continuación)

Atributo	Descripción
Precondiciones	1. El usuario debe utilizar un navegador web compatible. 2. El usuario debe encontrarse en la pantalla de bienvenida (`welcome.html`).
Flujo principal	1. Un usuario eligió el ”botón de login.[en] la primera página web. 2. El sistema crea un desafío de código que reenvía a la interfaz de Amasazon Cognito. 3. El usuario ingresa sus datos de acceso. 4. Si las credenciales ingresadas son correctas. 4.1 Amazon Cognito devuelve un código de autorización. 4.2 La aplicación intercambia el código por tokens JWT y los almacena localmente. 4.3 El sistema redirige al usuario a su panel correspondiente.
Postcondiciones	El usuario obtiene una sesión local válida y accede a las funcionalidades del sistema.
Flujos alternativos	1.1 Autenticación fallida: Credenciales Inválidas.

Como complemento a la operación base, en la tabla XLVI se detalla el flujo alternativo de excepción que el sistema ejecuta cuando las credenciales proporcionadas no son válidas.

Tabla XLVI: Flujo Alternativo 1.1: Credenciales Inválidas

Atributo	Descripción
Id	1.1
Breve descripción	Si existe error en el ingreso de credenciales el sistema niega el acceso.

Tabla XLVI: Flujo Alternativo 1.1 (Continuación)

Atributo	Descripción
Actores primarios	Estudiante, Administrador.
Precondiciones	1. Se evidencia el incorrecto ingreso de datos por parte del usuario.
Flujo alternativo	1. Inicia el flujo alternativo en el punto 4 del flujo principal. 2. Mediante Amazon Cognito se muestra mensaje que informa el error de credenciales. 3. Para reintentar el usuario permanece en la pantalla de inicio.
Postcondiciones	Ninguna.

3.14.2 CU-02: Registro de Auditoría de Accesos

Para garantizar la trazabilidad y seguridad en la plataforma, este caso de uso detalla el proceso automático de registro de sesiones. En la figura 4 se ilustra la arquitectura de este evento asíncrono.

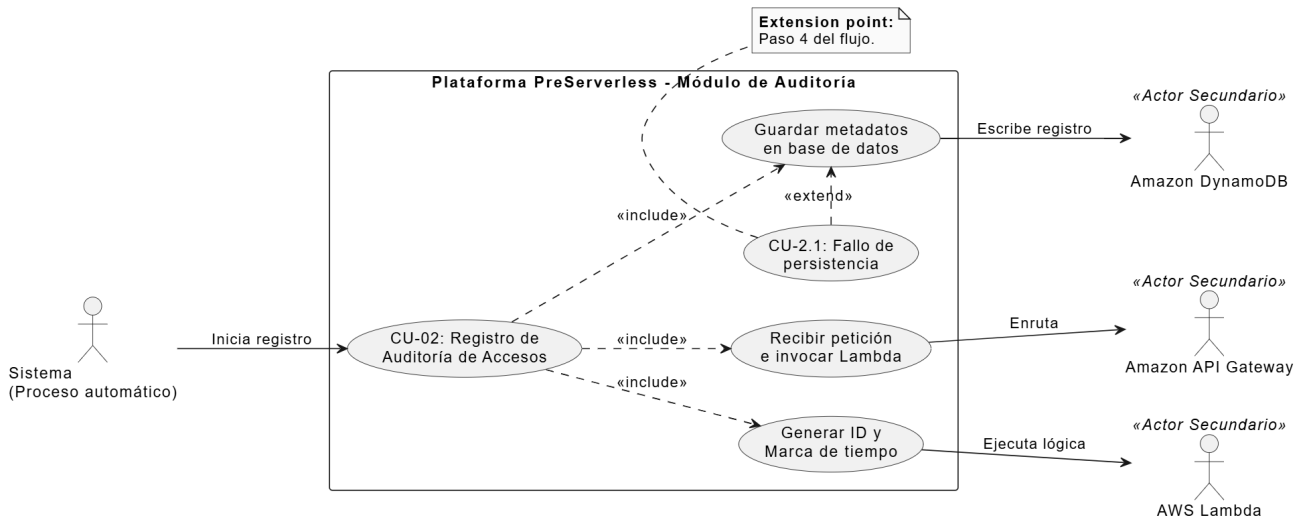


Figura 4.
Caso de uso Registro de Auditoría

Como se detalla en la tabla XLVII, el sistema captura y almacena los metadatos de conexión sin requerir intervención manual, delegando la carga de trabajo a los servicios en la nube.

Tabla XLVII: Especificación de CU-02: Registro de Auditoría de Accesos

Atributo	Descripción
Identificador	2
Breve descripción	Registra de forma automática las trazas de inicio y cierre de sesión.
Actores primarios	Sistema (Proceso automático).
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.
Precondiciones	1. El usuario ha iniciado o cerrado sesión.
Flujo principal	1. El frontend dispara una petición asíncrona con los datos de sesión. 2. Amazon API Gateway recibe la petición e invoca a AWS Lambda. 3. AWS Lambda genera un identificador único y la marca de tiempo. 4. Si hay conexión disponible con la base de datos. 4.1 AWS Lambda envía el comando para guardar los metadatos en Amazon DynamoDB. 4.2 AWS Lambda devuelve éxito al frontend.
Postcondiciones	Se guarda acción correctamente en la tabla logins.
Flujos alternativos	2.1 Fallo de persistencia en Base de Datos.

Dado que este proceso depende estrictamente de la disponibilidad de la base de datos, es crucial contemplar posibles interrupciones del servicio. La tabla XLVIII expone el comportamiento de tolerancia a fallos, priorizando siempre que la experiencia del usuario no se vea bloqueada por un error interno de auditoría.

Tabla XLVIII: Flujo Alternativo 2.1: Fallo de persistencia

Atributo	Descripción
Id	2.1
Breve descripción	El sistema omite el guardado debido a un error interno con la base de datos.
Actores primarios	Sistema (Proceso automático).
Precondiciones	1. Ha ocurrido un error de conexión al intentar guardar.
Flujo alternativo	1. El flujo alternativo comienza en el punto 4 del flujo principal. 2. AWS Lambda captura la excepción generada por Amazon DynamoDB. 3. AWS Lambda retorna una respuesta HTTP 200 con estado falso, evitando bloquear al usuario.
Postcondiciones	Ninguna.

3.14.3 CU-03: Selección de Área y Validación de Reglas

El paso previo a cualquier evaluación consiste en la configuración paramétrica de la prueba por parte del estudiante. En la figura 5 se esquematiza este proceso de selección y la validación de las reglas de negocio.

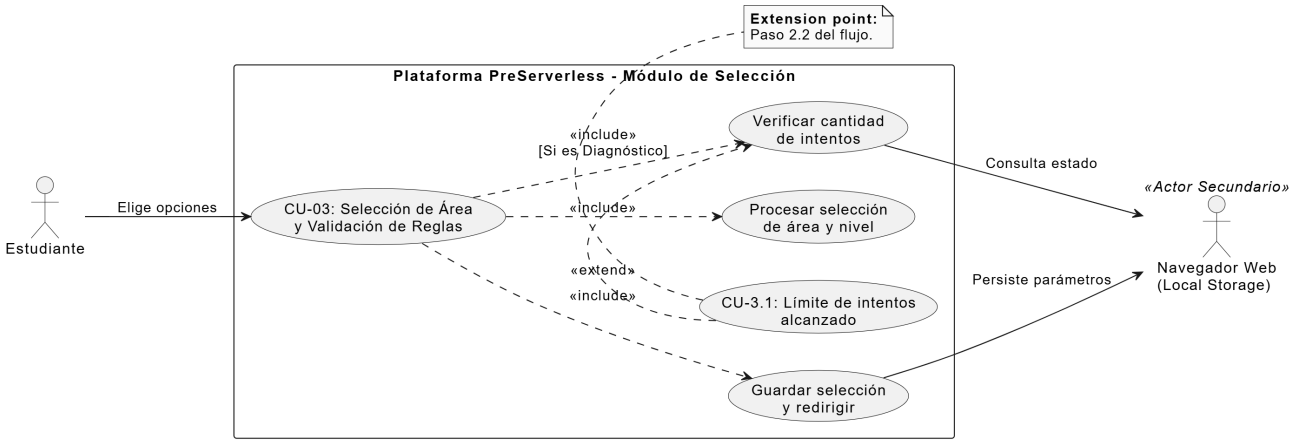


Figura 5.
Caso de uso Selección de Área

La especificación presentada en la tabla XLIX detalla el flujo principal donde se capturan las preferencias del usuario y se verifican los requisitos a nivel de cliente (almacenamiento local) antes de conceder el acceso al motor de evaluación.

Tabla XLIX: Especificación de CU-03: Selección de Área y Validación de Reglas

Atributo	Descripción
Identificador	3
Breve descripción	Permite elegir la categoría y dificultad, validando el límite de intentos.
Actores primarios	Estudiante.
Actores secundarios	Navegador Web (Local Storage).
Precondiciones	1. El estudiante se encuentra en la pantalla de selección (`seleccion.html`).

Tabla XLIX: Especificación de CU-03 (Continuación)

Atributo	Descripción
Flujo principal	1. El estudiante selecciona un área y un nivel de dificultad. 2. Si el nivel de dificultad seleccionado es ”Diagnóstico”. 2.1 El sistema verifica la cantidad de intentos consumidos en el almacenamiento local. 2.2 Extension point: Límite de intentos alcanzado. 3. El sistema guarda la selección temporalmente. 4. El sistema redirige a la pantalla de evaluación.
Postcondiciones	El estudiante ingresa a la vista de evaluación.
Flujos alternativos	3.1 Límite de intentos alcanzado.

Para mantener la integridad metodológica de la plataforma y evitar el abuso de las evaluaciones, el sistema impone restricciones de acceso temporales. Si el estudiante agota su cuota permitida, se ejecuta el escenario de bloqueo detallado en la tabla L.

Tabla L: Flujo Alternativo 3.1: Límite de intentos alcanzado

Atributo	Descripción
Id	3.1
Breve descripción	El sistema bloquea el acceso a la prueba diagnóstico por agotar intentos.
Actores primarios	Estudiante.
Precondiciones	1. El estudiante ha consumido sus 3 intentos y no ha pasado el tiempo de enfriamiento (1 hora).

Tabla L: Flujo Alternativo 3.1 (Continuación)

Atributo	Descripción
Flujo alternativo	1. El flujo alternativo comienza en el punto 2.2 del flujo principal. 2. El sistema lanza una alerta indicando los minutos faltantes para la renovación. 3. El estudiante permanece en la pantalla de selección.
Postcondiciones	Ninguna.

3.14.4 CU-04: Ejecución del Motor de Evaluación

El núcleo interactivo de la plataforma reside en el motor de evaluación, el cual recrea las condiciones de un examen real mediante la presentación dinámica de reactivos. En la figura 6 se ilustra la arquitectura de este proceso, destacando la interacción continua del usuario y la conexión con la base de datos.

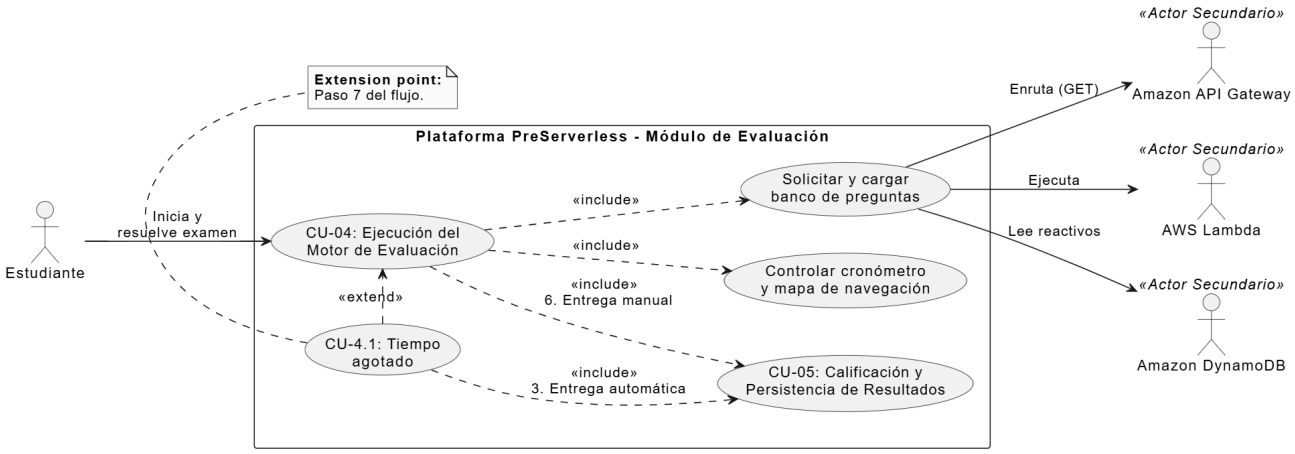


Figura 6.
Caso de uso Ejecución de Evaluación

La tabla LI especifica la secuencia de operaciones del flujo principal. Este abarca desde la extracción aleatoria de las preguntas almacenadas en la infraestructura de AWS, hasta la gestión del mapa de navegación visual y la entrega voluntaria del cuestionario por parte del estudiante.

Tabla LI: Especificación de CU-04: Ejecución del Motor de Evaluación

Atributo	Descripción
Identificador	4
Breve descripción	El estudiante resuelve preguntas dinámicas de opción múltiple, completación o emparejamiento con un cronómetro regresivo.
Actores primarios	Estudiante.
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.
Precondiciones	1. El estudiante posee intentos válidos y se encuentra en `prueba.html`.
Flujo principal	1. El sistema solicita el banco de preguntas a AWS Lambda. 2. AWS Lambda realiza consulta Amazon DynamoDB y retorna un máximo de 10 reactivos aleatorios. 3. Se inicia el cronometro del sistema regresivo de 600 segundos. 4. Mientras haya preguntas sin responder y el cronómetro sea mayor a cero. 4.1 El estudiante ingresa una respuesta y avanza. 4.2 El sistema actualiza el mapa de navegación visual. 5. El estudiante presiona el botón .Entregar Examenz confirma la acción. 6. Include (Calificación y Persistencia de Resultados). 7. Extension point: Tiempo agotado.
Postcondiciones	El examen finaliza y el estudiante es redirigido a la retroalimentación.
Flujos alternativos	4.1 Tiempo agotado.

Una de las restricciones más rigurosas durante la resolución de la prueba es el límite temporal. Para asegurar la imparcialidad y simular la presión académica, el sistema cuenta con un mecanismo de interrupción de estado. La tabla LII documenta este flujo alternativo, el cual toma el control y fuerza la

finalización del examen una vez que el cronómetro llega a cero.

Tabla LII: Flujo Alternativo 4.1: Tiempo agotado

Atributo	Descripción
Id	4.1
Breve descripción	Se radicaliza la finalización del sistema de examen al expirar el tiempo.
Actores primarios	Estudiante.
Precondiciones	1. El cronómetro regresivo llega a cero.
Flujo alternativo	1. El flujo alternativo comienza en el punto 7 del flujo principal. 2. El sistema interrumpe la interacción del estudiante. 3. Include (Calificación y Persistencia de Resultados).
Postcondiciones	El examen finaliza automáticamente.

Modelado Algorítmico de Diagrama de Actividades Dado que la arquitectura *Serverless* carece de un servidor de aplicaciones en ejecución continua, la lógica de control del examen y la gestión del estado se delegaron a una interacción estructurada entre el cliente (SPA) y los microservicios (AWS Lambda).

Con el propósito de formalizar a este algoritmo, en la Figura 7 se muestra el diagrama UML de Actividades perteneciente al motor de evaluación (CU-04). El presente modelo visualiza dos flujos de la ejecución concurrentes (fork). Por una parte, hay el Control Asíncrono del Cronómetro Regresivo y del otro hay el Bucle Iterativo en que resuelven reactivos los estudiantes. Esta bifurcación asegura una salida con seguridad para la prueba, así como envío seguro del payload a la nube en la mano de la entrega voluntaria o estrictamente bajo el timeout.

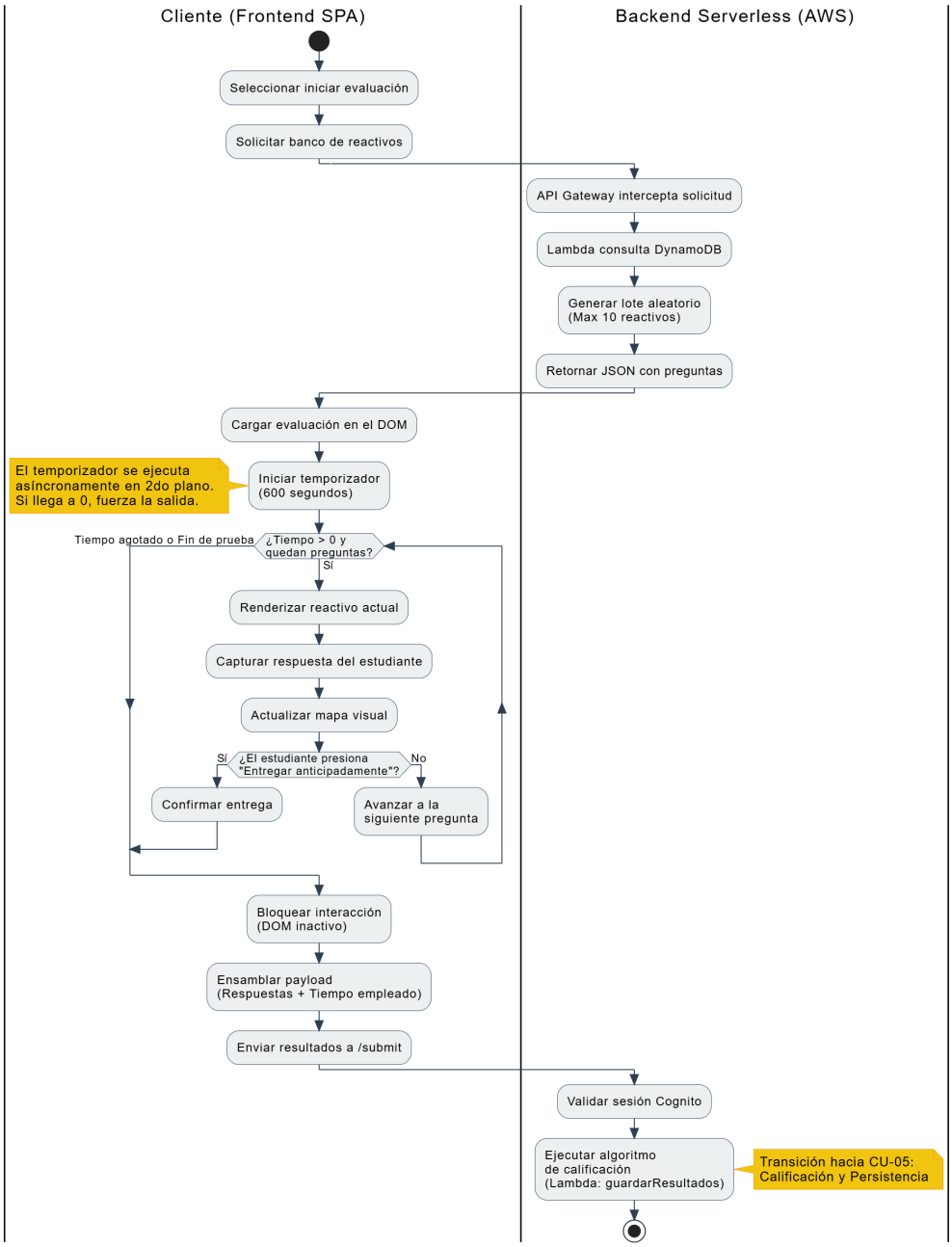


Figura 7.
Diagrama de Actividades UML: Lógica concurrente del motor de evaluación.

3.14.5 CU-05: Calificación y Persistencia de Resultados

Una vez concluida la evaluación, es imperativo procesar las métricas de rendimiento de manera inmediata para actualizar el progreso del estudiante y alimentar el motor de gamificación. En la figura 8 se esquematiza el proceso de calificación y su respectiva transmisión a la infraestructura en la nube.

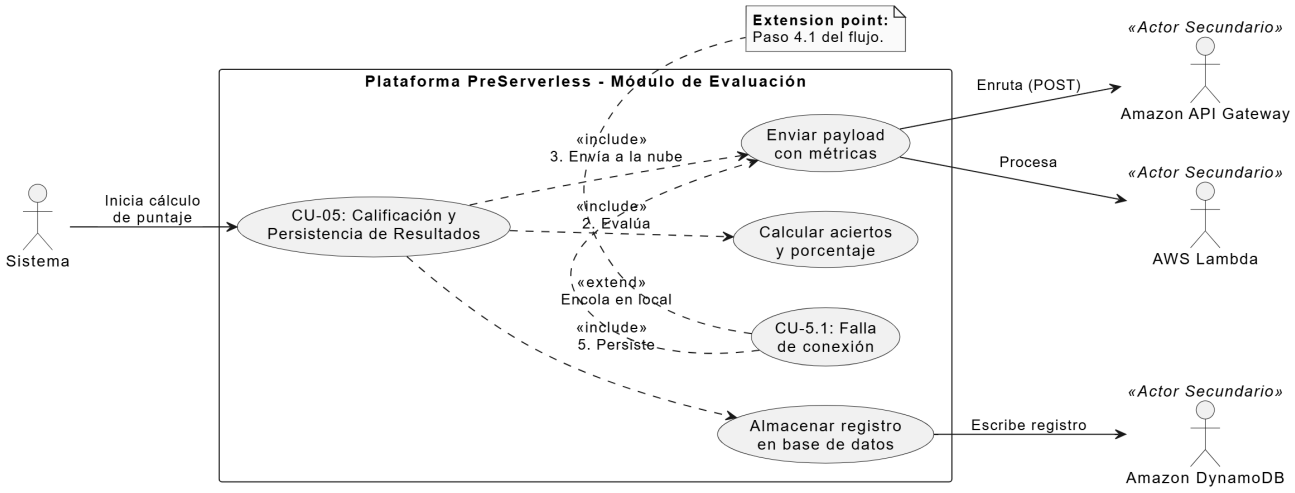


Figura 8.
Caso de uso Calificación de Resultados

Las operaciones algorítmicas que ejecuta el sistema se detallan en la tabla LIII. En este flujo vemos una alta eficiencia en el sentido de que después de la calificación, mueve todo el costo computacional hacia el lado de cliente (frontend). Luego se agrega/ensambla/transmitir el payload consolidado hacia los servicios de AWS para un almacenamiento permanente.

Tabla LIII: Especificación de CU-05: Calificación y Persistencia de Resultados

Atributo	**Descripción**
Identificador	5
Breve descripción	El sistema evalúa las respuestas, calcula el porcentaje de aciertos y persiste la métrica en la nube.
Actores primarios	Sistema (Frontend y Backend).
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.
Precondiciones	1. El estudiante ha entregado el examen de forma manual o automática.

Tabla LIII: Especificación de CU-05 (Continuación)

Atributo	Descripción
Flujo principal	1. El sistema (frontend) contrasta las respuestas ingresadas con la clave correcta del JSON en memoria. 2. El sistema calcula las veces que acertó, en porcentajes y también en tiempo usado. 3. . El sistema devuelve el payload junto con las métricas a AWS lambda via API Gateway. 4. Si ocurre un error de red al intentar enviar los datos. 4.1 Extension point: Conexión con falla. 5. AWS Lambda almacena el registro en Amazon DynamoDB. 6. Borrando progreso local en el sistema, redireccionado al modo vista de retroalimentación.
Postcondiciones	El puntaje queda guardado permanentemente para el ranking.
Flujos alternativos	5.1 Falla de conexión.

Dada la naturaleza de las aplicaciones web, es vital garantizar la resiliencia operativa frente a fluctuaciones de red o latencias prolongadas. La tabla LIV describe la estrategia de contingencia adoptada frente a estos escenarios, utilizando una cola de almacenamiento local para evitar cualquier pérdida de información académica crítica.

Tabla LIV: Flujo Alternativo 5.1: Falla de conexión

Atributo	Descripción
Id	5.1
Breve descripción	El sistema encola el resultado localmente para prevenir pérdida de datos.
Actores primarios	Sistema.

Tabla LIV: Flujo Alternativo 5.1 (Continuación)

Atributo	Descripción
Precondiciones	1. No hay conexión a internet o el servidor devuelve error.
Flujo alternativo	1. El flujo alternativo comienza en el punto 4.1 del flujo principal. 2. El frontend captura la excepción de red. 3. El payload es guardado en la cola del almacenamiento local (submitQueue) para que se pueda reenviar en el futuro. 4. La ejecución continúa en el punto 6 del flujo principal.
Postcondiciones	El puntaje queda en espera de sincronización.

3.14.6 CU-06: Clasificación Global Dinámica

El componente de gamificación de la plataforma se materializa a través de la tabla de posiciones o *Leaderboard*. En la figura 9 se detalla el proceso mediante el cual el estudiante consulta su rendimiento frente al resto de la comunidad.

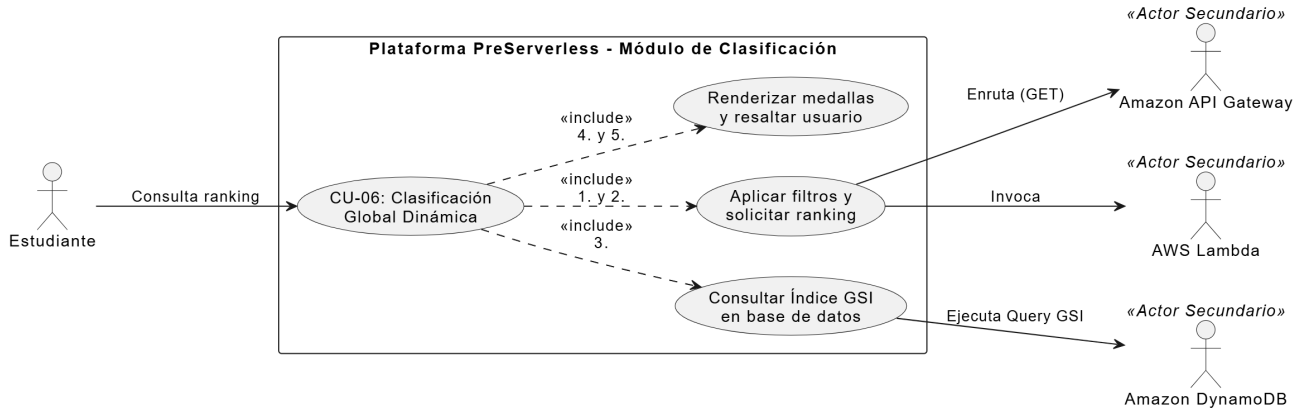


Figura 9.
Caso de uso Clasificación Global

La tabla LV expone la especificación de este flujo directo. A nivel arquitectónico, destaca el uso de un Índice Secundario Global (GSI) en Amazon DynamoDB para lograr una consulta y ordenamiento efi-

ciente de los puntajes, culminando con un renderizado dinámico que incentiva la competencia al resaltar al usuario en pantalla.

Tabla LV: Especificación de CU-06: Clasificación Global Dinámica

Atributo	**Descripción**
Identificador	6
Breve descripción	Consulta la base de datos para mostrar la tabla de posiciones (Leaderboard) asignando medallas y resaltando al usuario.
Actores primarios	Estudiante.
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.
Precondiciones	1. El estudiante ingresa a la vista de resultados (`resultado.html`).
Flujo principal	1. El estudiante selecciona los filtros de categoría y dificultad. 2. El sistema solicita el ranking a AWS Lambda. 3. AWS Lambda realiza un *Query* al Índice Secundario Global (GSI) ordenado descendentemente por puntaje en Amazon DynamoDB. 4. Para cada estudiante en el Top 100. 4.1 El sistema renderiza la fila, asignando medallas a los tres primeros lugares. 5. Si el correo de una fila coincide con el usuario autenticado. 5.1 El sistema aplica un estilo visual distintivo a la fila y añade la insignia ”TÚ”.
Postcondiciones	El usuario visualiza la tabla de posiciones actualizada.
Flujos alternativos	Ninguno.

3.14.7 CU-07: Monitoreo Analítico del Administrador

Para gestionar el impacto académico de la plataforma, el sistema provee herramientas de inteligencia de datos exclusivas para la gerencia. En la figura 10 se esquematiza el módulo de reportes y gráficas estadísticas.

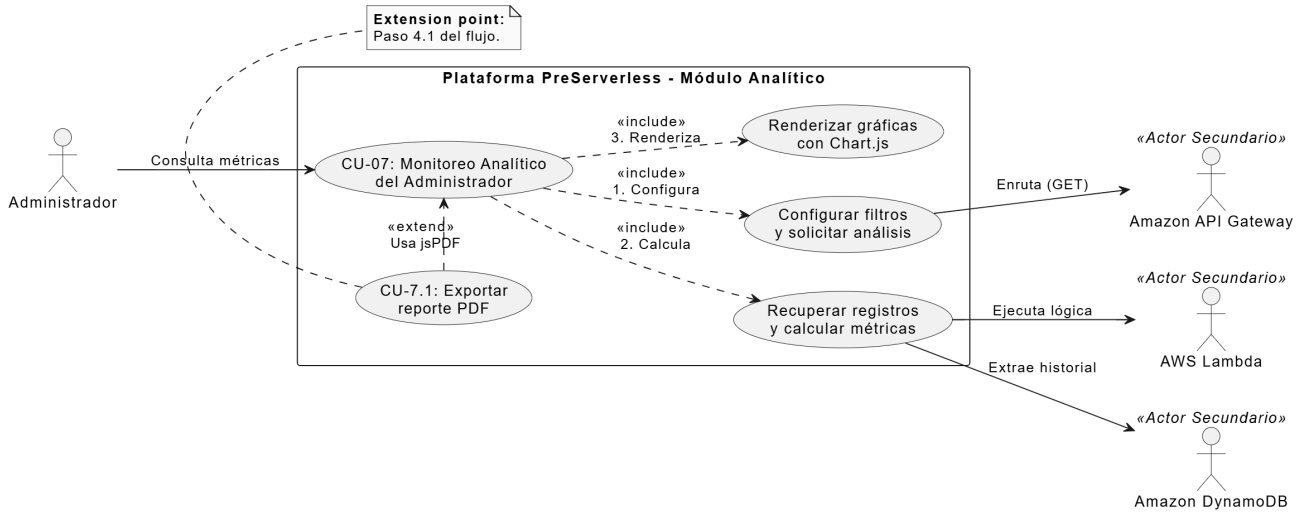


Figura 10.
Caso de uso Monitoreo Analítico

Como se detalla en la tabla LVI, el administrador tiene la capacidad de aplicar filtros cruzados para que AWS Lambda procese métricas avanzadas (como medidas de tendencia central y retención). Posteriormente, el frontend traduce esta información en visualizaciones interactivas de alto valor descriptivo.

Tabla LVI: Especificación de CU-07: Monitoreo Analítico del Administrador

Atributo	Descripción
Identificador	7
Breve descripción	Estadísticas descriptivas (mediana, moda, varianza) y gráficas de embudo, permiten realizar la evaluación del progreso de los estudiantes.
Actores primarios	Administrador.
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.

Tabla LVI: Especificación de CU-07 (Continuación)

Atributo	Descripción
Precondiciones	1. El administrador tiene sesión válida e ingresa a `admin.html`.
Flujo principal	1. Los filtros de fecha, categoría y dificultad, son configurados por el Administrador al pulsar Analizar. 2. Los registros de Amazon DynamoDB son recuperados por AWS Lambda y también calcula las métricas avanzadas y el embudo de retención. 3. Las gráficas son renderizadas de manera dinámica utilizando la librería en el sistema Chart.js. 4. Si el administrador desea conservar el análisis. 4.1 Extension point: Exportar PDF.
Postcondiciones	El administrador dispone de inteligencia de datos sobre la plataforma.
Flujos alternativos	7.1 Exportar PDF.

Adicional a la consulta visual en pantalla, el sistema contempla un requerimiento de documentación formal. La tabla LVII especifica el flujo de extensión que permite al administrador compilar todo el análisis renderizado en un archivo PDF descargable.

Tabla LVII: Flujo Alternativo 7.1: Exportar PDF

Atributo	Descripción
Id	7.1
Breve descripción	Se presenta en pantalla un documento PDF generado de manera estructurada.
Actores primarios	Administrador.

Tabla LVII: Flujo Alternativo 7.1 (Continuación)

Atributo	Descripción
Precondiciones	1. Se completa el análisis de datos y la renderización de los gráficos.
Flujo alternativo	1. El flujo alternativo comienza en el punto 4.1 del flujo principal. 2. El administrador presiona el botón .Exportar PDF". 3. El sistema procesa los componentes del DOM y los gráficos de Canvas. 4. El sistema utiliza la librería *jsPDF* para construir y descargar el documento en el dispositivo del usuario.
Postcondiciones	Se descarga el archivo PDF con el reporte.

3.14.8 CU-08: Gestión de Reactivos vía JSON

Para escalar la plataforma sin depender de inserciones manuales tediosas, se diseñó un módulo de carga masiva de contenido. En la figura 11 se modela el proceso de alimentación de la base de datos de preguntas.

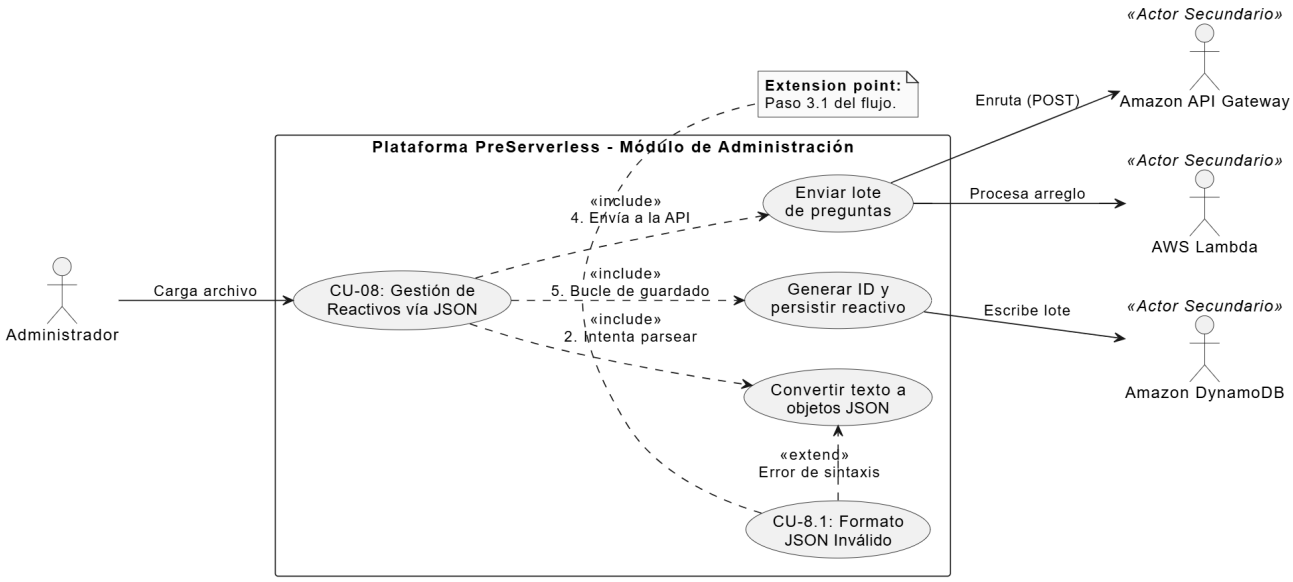


Figura 11.
Caso de uso Gestión de Reactivos

La especificación de la tabla LVIII detalla la secuencia de ingesta de datos. El flujo aprovecha la capa- cidad de las funciones Serverless (AWS Lambda) para iterar sobre arreglos y persistir múltiples registros de forma secuencial o paralela en DynamoDB, notificando al usuario una vez concluido el lote.

Tabla LVIII: Especificación de CU-08: Gestión de Reactivos vía JSON

Atributo	Descripción
Identificador	8
Breve descripción	Permite poblar la base de datos de preguntas subiendo un archivo estructurado en formato JSON.
Actores primarios	Administrador.
Actores secundarios	Amazon API Gateway, AWS Lambda, Amazon DynamoDB.
Precondiciones	1. El administrador se encuentra en la sección de carga masiva.
Flujo principal	1. El administrador carga un archivo de texto con formato JSON o pega el texto directamente. 2. El sistema intenta convertir el texto a objetos de JavaScript. 3. Si el formato JSON está mal estructurado o contiene errores de sintaxis. 3.1 Extension point: JSON Inválido. 4. El sistema envía el arreglo de preguntas a AWS Lambda. 5. Para cada reactivo en el lote enviado. 5.1 AWS Lambda genera un identificador único y guarda el ítem en Amazon DynamoDB. 6. El sistema notifica al administrador la cantidad total de preguntas insertadas exitosamente.
Postcondiciones	Las nuevas preguntas quedan disponibles para los exámenes.
Flujos alternativos	8.1 JSON Inválido.

Dado que la carga masiva requiere un formato de datos estricto, el sistema incluye una capa de validación temprana. La tabla LIX describe el comportamiento de interrupción local frente a sintaxis inválida, previniendo el envío de peticiones defectuosas a la nube y optimizando los costos de la arquitectura Serverless.

Tabla LIX: Flujo Alternativo 8.1: JSON Inválido

Atributo	Descripción
Id	8.1
Breve descripción	Detiene la carga debido a un formato de datos ilegible.
Actores primarios	Administrador.
Precondiciones	1. El estándar de notación JSON, no es cumplido por el texto ingresado.
Flujo alternativo	1. El flujo alternativo arranca en el punto 3.1 tomando en cuenta el flujo principal. 2. En el frontend, la función parceo genera un error. 3. Se muestra una alerta roja en el sistema que indica que el administrador necesita comillas faltantes o saltos de línea.
Postcondiciones	La base de datos no recibe información y la operación se aborta.

3.15 Modelado Dinámico: Diagramas de Secuencia

Con el fin de entender la interacción asíncrona de los componentes de la arquitectura Serverless se construyeron diagramas de secuencia para los flujos críticos del sistema.

3.15.1 Secuencia de Evaluación y Persistencia

El primer diagrama detalla el flujo de calificación. El cliente (Estudiante) envía un *payload* JSON mediante el método POST a la ruta `/submit`. API Gateway intercepta la petición y valida el token JWT contra Amazon Cognito. Tras la autorización exitosa, la Lambda `guardarResultados` ejecuta la comparación algorítmica y realiza una operación `PutItem` en la tabla `resultados` de DynamoDB, retornando finalmente un código HTTP 201 al cliente.

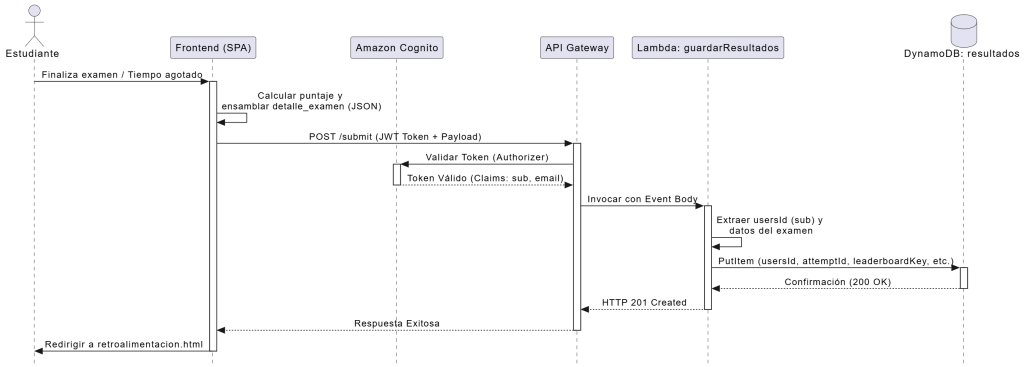


Figura 12.
Diagrama de secuencia: Ejecución y calificación del simulacro.

3.15.2 Secuencia de Auditoría de Seguridad

Se garantiza que el segundo flujo tenga trazabilidad del sistema. Cognito devuelve el IdToken después del login (flujo PKCE). Una vez hecho esto, el frontend hace un request asíncrono de peticion a /log-login. Lambda saveloginlog capta el sub del usuario, la ip, la timestamp, e insertar un log a la tabla de logins. Dado que no es una operación bloqueante, el estudiante accedió al dashboard y vio cero latencias en la red.

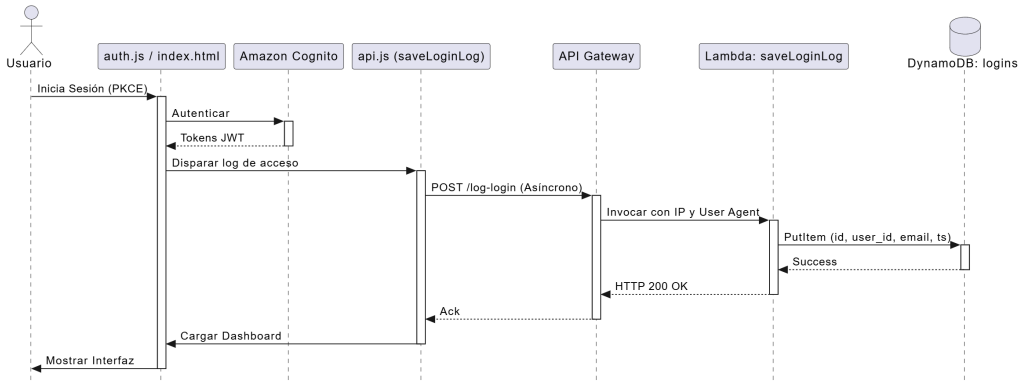


Figura 13.
Diagrama de secuencia: Registro de trazabilidad y accesos.

3.16 Modelado Estructural: Modelo Lógico NoSQL

A diferencia de las bases de datos relacionales, el modelado estructural en DynamoDB no se representa con relaciones foráneas, sino mediante el diseño de los documentos almacenados. La Figura 14 ilustra el modelo lógico de las tres tablas exclusivas que operan en el sistema.

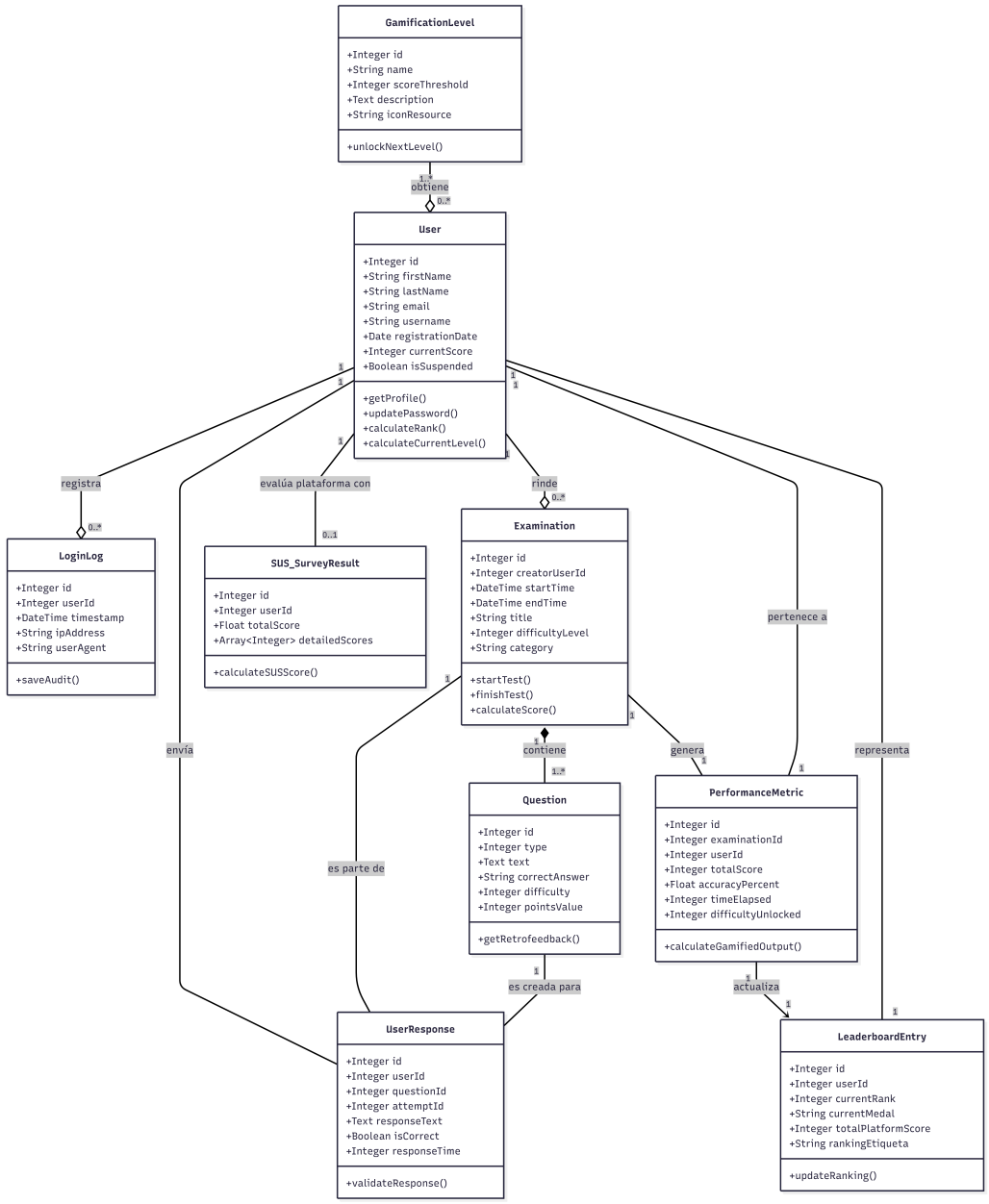


Figura 14.
Modelo lógico de datos en Amazon DynamoDB.

A efectos de correcta interpretación del modelo de persistencia, se detallan las tres únicas entidades implementadas:

- **`questions`:** Almacena el banco de reactivos. Usa id como clave de particionamiento (PK). Soporta múltiples atributos polimórficos para tipos de pregunta distintos (mcq, completion).
- **`resultados`:** Almacena el historial de intentos. Su clave de partiticionado es el userId y la clave de ordernación (SK) es el Timestamp. Incluye un índice secundario global (GSI) en el puntaje para

generar el Leaderboard.

- `logins:` Tabla de seguridad y trazabilidad registra cada acceso utilizando userId Como PK y Capturing Network Metadat (IP, Browser) en el sistema.

3.17 Ejecución Ágil: Adaptación, Roles y Ciclo de Vida

En la industria del software, el marco Scrum requiere de un equipo multidisciplinario con dedicación exclusiva. Sin embargo, para la ejecución de un trabajo de integración curricular, resulta indispensable realizar una adaptación metodológica. En este proyecto, se aplicó una **adaptación ligera** de Scrum, la cual conserva intactos los principios empíricos de iteración, inspección y transparencia, apoyándose en la evidencia académica sobre la efectividad de Scrum en proyectos de desarrollo individual [47].

3.17.1 Estructura del Equipo y Asignación de Roles

Para garantizar la validez académica y el rigor de la ingeniería del software, los roles de inspección, validación y desarrollo fueron distribuidos entre las autoridades de la Universidad Técnica del Norte y el autor de la investigación. La Tabla LX detalla esta asignación formal.

Tabla LX: Roles de Scrum adaptados a la estructura del proyecto

Rol en Scrum	Responsable	Responsabilidades en el proyecto
Product Owner	Rafael G. Lara	Definición de la visión y alcance académico, priorización de funcionalidades orientadas a la pedagogía y validación/aprobación de los incrementos de software.
Scrum Master / Development Team	Rafael G. Lara	Autogestión metodológica, diseño de la arquitectura en la nube (*FaaS*), programación del *frontend* y *backend*, gestión de bases de datos NoSQL, pruebas y redacción documental.

Asesores Técnicos	MSc. Fausto Salazar, PhD y MSc. Pablo Landeta	Consultoría especializada en calidad de código, revisión de la viabilidad de la infraestructura propuesta y auditoría técnica puntual.

3.17.2 Matriz de Responsabilidades (RACI)

Para establecer con absoluta claridad la rendición de cuentas sobre los componentes críticos de la arquitectura *Serverless* y la documentación de la tesis, se estructuró una Matriz RACI: **R**esponsable (*Responsible*), **A**probador (*Accountable*), **C**onsultado (*Consulted*) e **I**nformado (*Informed*).

Esta matriz (Tabla LXI) demuestra que, si bien la codificación recae en un solo individuo, el proceso de validación y auditoría involucra un flujo de revisión institucional [44].

Tabla LXI: Matriz RACI para actividades clave del ciclo de vida del software

Actividad / Componente	**R**	**A**	**C**	**I**
Análisis y diseño de la arquitectura FaaS	Rafael	Tutor	Asesor	–
Despliegue de Frontend (Amazon S3)	Rafael	–	Tutor	–
Programación de microservicios (AWS Lambda)	Rafael	Tutor	Asesor	–
Estructuración y persistencia (DynamoDB)	Rafael	–	Tutor	–
Pruebas piloto y levantamiento de datos (SUS)	Rafael	Tutor	–	–
Documentación, redacción y edición de tesis	Rafael	Tutor	Asesor	–

3.17.3 Fases del Proyecto y Cronograma Macro

El proyecto de titulación se planificó sobre un horizonte temporal de 18 meses. Para sincronizar orgánicamente el desarrollo iterativo del código fuente con la redacción de la memoria técnica, el ciclo de vida global se dividió en tres fases macro, como se expone en la Tabla LXII.

Tabla LXII: Cronograma Macro de Fases de Desarrollo y Redacción

Fase	Periodo	Hitos de Desarrollo y Documentación
Fase I: Fundamentación y Diseño	Marzo 2025 – Agosto 2025	Definición del alcance del proyecto, levantamiento de requerimientos pedagógicos, diseño de la interfaz y redacción teórica de los Capítulos I y II (Marco Teórico).
Fase II: Ejecución Ágil y Validación	Septiembre 2025 – Marzo 2026	Desarrollo de sprints: Ejecución de los Sprints de desarrollo; Programación de las funciones de AWS Lambda; Conexión con DynamoDB; Deploy del pilote con 50 usuarios reales; Redacción metodológica de los Capítulos III y IV.
Fase III: Refinamiento y Cierre	Abril 2026 – Agosto 2026	Monitoreo de estabilidad, corrección de incidentes menores de usabilidad, refinamiento del diseño GUI y revisión editorial final del documento de grado.

De estas tres fases, la **Fase II** constituye el núcleo técnico del trabajo, en la cual se aplicaron estrictamente los *Sprints* de Scrum y los principios de gamificación (MDA) para la construcción tangible de la plataforma *Pre Serverless*.

3.17.4 Registro de Eventos Scrum y Tutorías Académicas

La adaptación del marco Scrum a un entorno de desarrollo individual requiere un rigor especial en la ejecución de los eventos de inspección y adaptación. Los eventos estandarizados (Planificación, Sincronización, Revisión y Retrospectiva) se fusionaron de manera estratégica con las tutorías académicas oficiales y los puntos de control (*Checkpoints*) establecidos por el Director de Tesis.

Un registro formal de estas interacciones asegura la transparencia del proceso iterativo. La evolución de la arquitectura Sin Servidor está evidenciada por una retroalimentación continua. La Tabla LXIII compila los eventos importantes que se llevaron a cabo en la Fase II del Proyecto, resaltando las decisiones técnicas y pedagógicas.

Tabla LXIII: Registro de eventos iterativos, revisiones y tutorías académicas

Sprint	Mes / Año	Evento Académico	Resumen y Decisiones Técnicas
S1	Septiembre 2025	*Sprint Planning*	**Objetivo:** Despliegue del núcleo S3 y Amazon Cognito. **Decisión:** Se establece el estándar CORS a fin de comunicar de forma segura con API Gateway.
S2	Octubre 2025	*Checkpoint* Técnico	**Objetivo:** Auditoría de sesión. **Decisión:** Ajuste en el flujo PKCE; la Lambda `saveLoginLog` capturar la marca temporal (*timestamp*) nativa de AWS.
S3	Noviembre 2025	Tutoría de Avance	**Objetivo:** Validación del motor de examen multiformato. **Decisión:** Se aprobó la lógica de retroalimentación inmediata en el *frontend* sin recargar la página.

Continúa en la siguiente página

Tabla LXIII (continuación). Registro de eventos iterativos, revisiones y tutorías académicas

Sprint	Mes / Año	Evento Académico	Resumen y Decisiones Técnicas
S4	Diciembre 2025	*Sprint Retrospective*	**Objetivo:** Optimización del código fuente. **Acción:** Refactorización de las expresiones regulares en el calificador de ítems de completación.
S5	Febrero 2026	*Sprint Review*	**Objetivo:** Sistema de Gamificación operativo. **Decisión:** El Leaderboard global solo destaca visualmente el Top 3 (medallas) haciendo uso de consultas al Índice GSI.
S6	Marzo 2026	Tutoría de Tesis	**Objetivo:** Confirmación de alcance de la aplicación. **Decisión:** Se prioriza estabilizar el módulo de gráficas (*Chart.js*) antes de lanzar el piloto en producción.
S7	Marzo 2026	*Checkpoint* Piloto	**Objetivo:** Monitoreo del despliegue con 50 estudiantes. **Acción:** Resolución de concurrencia en Amazon Cognito para estabilizar accesos simultáneos.
S8	Finales Mar. 2026	Revisión Final	**Objetivo:** Cierre de la validación empírica. **Decisión:** Aprobación de los resultados de usabilidad (SUS) y cierre de la fase de desarrollo técnico.

Elaboración propia.

Por medio de estos sucesos, el rol del ”Product Owner”(Director de la tesis), permaneció vivo: Ca-

da incremento técnico creado bajo AWS tuvo respuesta directa a una necesidad de diseño pedagógico gamificado.

3.17.5 Artefactos de Scrum: Product Backlog Priorizado

El *Product Backlog* es el artefacto central de planificación en Scrum. Constituye la lista maestra, emergente y priorizada de todo lo que se conoce que es necesario para el producto [44]. En lugar de redactar especificaciones técnicas rígidas que limiten la creatividad del diseño, los requerimientos de la plataforma *Pre Serverless* se estructuraron bajo el formato estándar de **Historias de Usuario (HU)**: *Çomo [rol], quiero [funcionalidad] para [beneficio]"*.

Para garantizar la trazabilidad arquitectónica, el Product Owner (Director de Tesis) y el Development Team (Autor) han declarado que cada Historia del Usuario debe incluir criterios de aceptación técnicos enfocado en una infraestructura Serverless. La estimación del esfuerzo fue hecha usando Story Point basado en la complejidad del lógica de programa requerido para el AWS Lambda y las Reglas del Frontend (S3).

Para garantizar un control de calidad riguroso para cada iteración y certificarse de que los incrementos resultaran operativos en la nube, el equipo definieron una estricta Definición de "hecho". Una Historia de Usuario sólo se consideró una vez finalizada (Done en el Kanban), cuando cumplía con tres condiciones: 1.) El despliegue exitoso de la lógica del microservicio (AWS Lambda). 2) Ni siquiera mostró errores de ejecución o los problemas de CORS durante las pruebas de integración vía API Gateway, y 3) El cumplimiento de los criterios técnicos de la aceptación.

La Tabla LXIV consolida el "Product Backlog"definitivo que regía la Fase II del Proyecto. Las funcionalidades críticas de autenticación y evaluación se priorizaron y las características de refinación estética y medición empíricas fueron dejadas para el final del ciclo.

Tabla LXIV: Product Backlog Priorizado con Criterios de Aceptación Serverless

ID	Historia de Usuario	Criterios de Aceptación Técnicos	Prior.	Pts
HU-01	Como estudiante, quiero registrarme de forma segura para guardar mi puntaje.	Flujo PKCE exitoso vía Amazon Cognito; Token JWT inyectado en cabecera; persistencia de auditoría en DynamoDB.	Alta	3
HU-02	Como estudiante, quiero responder preguntas de opción múltiple para evaluar mi nivel.	Lambda `obtenerPreguntas` retorna 4 opciones; solo 1 es correcta; *feedback* inmediato en el DOM; puntaje acumulado en memoria.	Alta	5
HU-03	Como estudiante, quiero preguntas de completación (texto libre) para demostrar dominio.	Algoritmo en backend normaliza tildes y espacios; acepta cadenas equivalentes; califica precisión con Regex.	Alta	5
HU-04	Como estudiante, quiero ejercicios de unión de columnas (*matching*) para asociar conceptos.	Motor JS soporta *Drag & Drop*; función Lambda realiza calificación proporcional por emparejamientos correctos.	Alta	5
HU-05	Como estudiante, quiero desbloquear niveles de dificultad según mi rendimiento	El sistema habilita el nivel Normal o Difícil de acuerdo con el resultado obtenido en la evaluación, manteniendo la persistencia del estado del usuario.	Alta	3

Continúa en la siguiente página

Tabla LXIV (continuación). Product Backlog Priorizado con Criterios de Aceptación Serverless

ID	Historia de Usuario	Criterios de Aceptación Técnicos	Prior.	Pts
HU-06	Como estudiante, quiero visualizar mi puntaje durante la evaluación.	El Frontend actualiza el puntaje durante la evaluación sin recargar la página.	Media	3
HU-07	Como estudiante, quiero ver mi posición global en el *Leaderboard*.	Lambda `getLeaderboard` consulta el Índice Secundario Global (GSI) de DynamoDB; ordenamiento descendente por puntaje.	Media	5
HU-08	Como estudiante, quiero recibir medallas si alcanzo los primeros lugares del ranking.	El *frontend* procesa el arreglo del ranking y renderiza componentes visuales (Oro, Plata, Bronce) estrictamente a los índices 0, 1 y 2.	Media	3
HU-09	Como administrador, quiero ver el progreso y métricas de error de la plataforma.	Lambda `getMetrics` realiza agregación de datos; *Chart.js* renderiza % completado y cuellos de botella.	Media	3
HU-10	Como equipo, quiero medir la usabilidad con el instrumento SUS para validar calidad.	Despliegue de piloto a 50 usuarios; captura de datos de escala Likert; cálculo matemático de usabilidad (0-100).	Alta	5

Elaboración propia.

3.17.6 Especificación de Historias de Usuario y Criterios de Aceptación

Se trabaja en un desglose detallado para asegurar la rastreabilidad entre requerimientos funcionales e implementación técnica en Amazon Web Services (AWS). Estas fichas técnicas delimitan el alcance, la prioridad y los criterios de aceptación con que se gobierna la validación de los incrementos de software..

Gestión del control de acceso e identidades es el primero de los pilares de seguridad del sistema. A continuación, en la tabla que sigue se proporcionan detalles de los requisitos de implementación del flujo de autenticación delegada y el registro de auditoría.

Tabla LXV: Detalle de HU-01: Autenticación Segura y Auditoría

Atributo	**Descripción**
Título	Autenticación Segura y Gestión de Identidad.
Descripción	Como estudiante, quiero registrarme de forma segura para guardar mi puntaje.
Prioridad / Riesgo	Alta / Medio.
Estimación (Pts)	3 Puntos de Historia.
Criterios de Aceptación	1. El script auth.js debe resolver el flujo PKCE conectando con Amazon Cognito. 2. Se debe ejecutar la persistencia asíncrona de metadatos vía el microservicio saveLoginLog.mjs. 3. Inyección y validación del token JWT en el almacenamiento de sesión del cliente.

La evaluación dinámica es fundamental para evitar el sesgo de memorización en los aspirantes mediante el barajado aleatorio. La siguiente especificación describe el funcionamiento técnico del motor de preguntas de opción múltiple.

Tabla LXVI: Detalle de HU-02: Generación Dinámica de Reactivos (MCQ)

Atributo	Descripción
Título	Generación Dinámica de Reactivos de Opción Múltiple.
Descripción	Como estudiante, quiero responder preguntas de opción múltiple para evaluar mi nivel sin patrones de memorización.
Prioridad / Riesgo	Alta / Bajo.
Estimación (Pts)	5 Puntos de Historia.
Criterios de Aceptación	1. La función obtenerPreguntas.mjs debe aplicar el algoritmo shuffle al arreglo de opciones. 2. El DOM debe renderizar el componente validando una única respuesta correcta frente a la clave en base de datos. 3. Generación de retroalimentación visual extrayendo el campo explicación desde DynamoDB.

La precisión terminológica se evalúa mediante reactivos que requieren el ingreso de texto por parte del usuario. Los criterios para la validación y normalización de estas entradas para evitar errores por tildes o mayúsculas se describen en la ficha subsecuente.

Tabla LXVII: Detalle de HU-03: Motor de Completación

Atributo	Descripción
Título	Evaluación de Completación con Normalización de Texto.
Descripción	Como estudiante, quiero preguntas de texto libre para demostrar dominio sin penalización por tildes o mayúsculas.
Prioridad / Riesgo	Alta / Medio.

Estimación (Pts)	5 Puntos de Historia.
Criterios de Aceptación	1. Uso de expresiones regulares en juegos.js para normalizar la entrada del usuario. 2. Aceptación de cadenas equivalentes extraídas del atributo correcta del JSON. 3. Se bloquea los campos de entrada luego del envío con el fin de evitar mutaciones de estado.

Asociación de conceptos clave: es posible hacer la asociación por medio de una interfaz en dónde se emparejan elementos. Los requisitos para la calificación proporcional y la gestión de este tipo de reactivos se pueden ver en la siguiente tabla:

Tabla LXVIII: Detalle de HU-04: Motor de Emparejamiento (Matching)

Atributo	**Descripción**
Título	Motor de Emparejamiento de Columnas.
Descripción	Como estudiante, quiero ejercicios de unión de columnas para asociar conceptos clave.
Prioridad / Riesgo	Alta / Medio.
Estimación (Pts)	5 Puntos de Historia.
Criterios de Aceptación	1. Implementación de lógica de arrastrar y soltar o selección encadenada en el cliente web. 2. Algoritmo de calificación iterativa calculando aciertos proporcionales. 3. Mapeo estructural del atributo parejas inyectado desde la base de datos.

Con el propósito de regular el flujo pedagógico y simular restricciones de acceso reales, el sistema implementa reglas de negocio para los intentos. La siguiente tabla detalla la lógica de bloqueo y renovación

temporal de pruebas.

Tabla LXIX: Detalle de HU-05: Progresión y Bloqueo de Niveles

Atributo	Descripción
Título	Control de Progresión y Desbloqueo Adaptativo.
Descripción	Como estudiante, quiero que el sistema adapte mi acceso a niveles superiores según mi rendimiento previo.
Prioridad / Riesgo	Media / Bajo.
Estimación (Pts)	3 Puntos de Historia.
Criterios de Aceptación	1. Control de estado en estado.js gestionado a través de almacenamiento local. 2. Restricción lógica basada en el consumo máximo de tres intentos en pruebas de diagnóstico. 3. Implementación algorítmica para renovar intentos tras el cumplimiento de sesenta minutos.

La interfaz del usuario durante la evaluación debe proporcionar información constante sobre el progreso y el tiempo restante. Los parámetros técnicos para el renderizado del mapa de navegación y el temporizador se presentan a continuación.

Tabla LXX: Detalle de HU-06: Interfaz HUD y Temporizador

Atributo	Descripción
Título	Interfaz Gráfica de Usuario y Temporizador Global.
Descripción	Como estudiante, quiero ver mi tiempo y avance de reactivos en tiempo real para gestionar la prueba.
Prioridad / Riesgo	Media / Bajo.

Estimación (Pts)	3 Puntos de Historia.
Criterios de Aceptación	1. Renderizado asíncrono del mapa de navegación sin originar recarga de la página web. 2. Se instala el temporizador regresivo global el mismo que activa la interrupción de estado al llegar a cero. 3. Renderizado del puntaje obtenido y la retroalimentación inmediata de las respuestas.

El componente competitivo de la gamificación requiere una consulta eficiente de los puntajes de toda la comunidad estudiantil. La tabla presentada a continuación describe la implementación del escalafón global utilizando índices secundarios.

Tabla LXXI: Detalle de HU-07: Clasificación Global (Leaderboard)

Atributo	Descripción
Título	Clasificación Global y Gamificación Territorial.
Descripción	Como estudiante, quiero ver mi posición en el ranking general para comparar mi desempeño con la comunidad.
Prioridad / Riesgo	Media / Medio.
Estimación (Pts)	5 Puntos de Historia.
Criterios de Aceptación	1. La función getLeaderboard.mjs debe ejecutar una consulta empleando el índice secundario global de DynamoDB. 2. Límite de consulta aplicado desde el backend para evitar sobrecarga de red y optimizar lecturas. 3. Ejecución de lógica visual en rankings.js para resaltar la fila del usuario con sesión activa.

Para fomentar la motivación intrínseca, el sistema otorga reconocimientos visuales automáticos basados en el rendimiento. En la siguiente ficha se detallan los criterios para la asignación de medallas y etiquetas de estatus académico.

Tabla LXXII: Detalle de HU-08: Sistema de Medallas y Estatus

Atributo	Descripción
Título	Sistema de Medallas y Estatus Académico.
Descripción	Como estudiante, quiero recibir condecoraciones visuales si alcanzo los primeros puestos del ranking.
Prioridad / Riesgo	Media / Bajo.
Estimación (Pts)	3 Puntos de Historia.
Criterios de Aceptación	1. Procesamiento estricto de los tres primeros índices del arreglo de resultados general. 2. Inserción dinámica en el DOM de componentes de medalla de oro, plata y bronce. 3. Renderizado de insignias de nivel basadas en la condicional de precisión porcentual del estudiante.

La gestión administrativa de la plataforma depende de la capacidad de visualizar métricas agregadas sobre el desempeño del banco de preguntas. Los requisitos para el panel analítico y el filtro de seguridad por roles se describen a continuación.

Tabla LXXIII: Detalle de HU-09: Panel Analítico del Administrador

Atributo	Descripción
Título	Monitoreo Analítico del Administrador.

Descripción	El administrador, desea visualizar métricas y posibles cuellos de botella para el mejoramiento del banco de preguntas
Prioridad / Riesgo	Media / Medio.
Estimación (Pts)	3 Puntos de Historia.
Criterios de Aceptación	1. El backend sólo tiene que autorizar el acceso cuando el token tenga el rol de administrador. 2. Entrega Top de Errores, Embudo Retención, Procesos Matriciales de Datos por cada departamento. 3. Renderización interactiva con consumo de librería gráfica chart.js en vista de administración.

Finalmente, el cierre del ciclo metodológico requiere la validación científica de la plataforma con usuarios reales. En la siguiente tabla se presentan los parámetros técnicos para la ejecución del piloto y la recolección de datos de usabilidad.

Tabla LXXIV: Detalle de HU-10: Validación Científica de Usabilidad

Atributo	**Descripción**
Título	Medición Empírica de Usabilidad y Consistencia.
Descripción	Como equipo, quiero medir matemáticamente la facilidad de uso del entorno gamificado en producción.
Prioridad / Riesgo	Alta / Alto.
Estimación (Pts)	5 Puntos de Historia.

Criterios de Aceptación	1. Aportar la ejecución del despliegue de piloto a una muestra controlada. Entorno Web: Producción. 2. Acopio fidedigno de las respuestas por medio de la escala Likert del aparato SUS. 3. Calculando métricas de consistencia interna y varianza de forma automatizada en Python.

3.17.7 Gestión Visual del Backlog

Para mantener la transparencia del trabajo (uno de los pilares empíricos de Scrum) y gestionar el flujo de las historias de usuario desde su concepción hasta su despliegue en la nube, se hizo uso de un tablero Kanban digital. La Figura 15 evidencia el entorno de trabajo donde se controló el ciclo de vida de cada ítem, desplazándolos por los estados de *To Do* (Por Hacer), *In Progress* (En Progreso), *Testing* (Pruebas Locales) y *Done* (Desplegado en AWS).

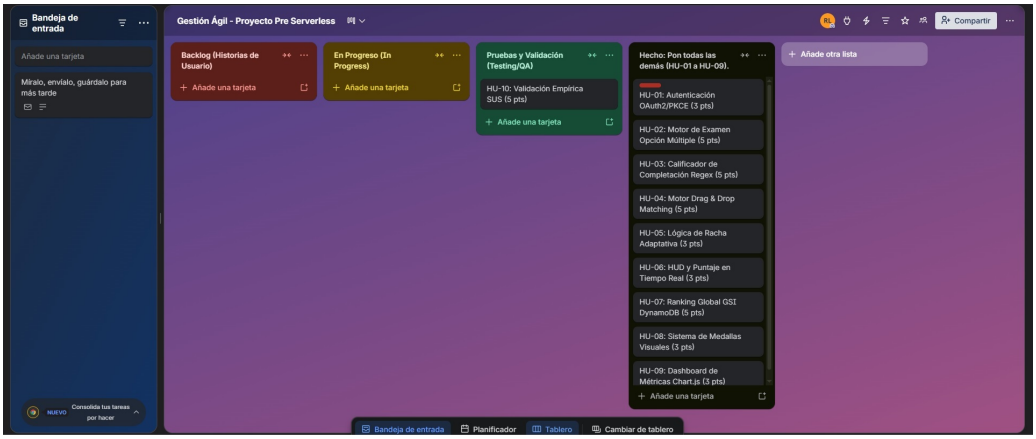


Figura 15.
Tablero Kanban utilizado para la gestión visual del Product Backlog.

3.17.8 Sprint Backlog y Ciclo de Desarrollo Técnico

Para materializar el *Product Backlog* durante la **Fase II** del proyecto (Septiembre 2025 – Marzo 2026), el desarrollo técnico se dividió en **9 Sprints**. Dado el carácter unipersonal del equipo de desarrollo,

la duración de cada iteración se adaptó de manera flexible (entre 3 y 4 semanas), asegurando que cada *Sprint* culminara con un incremento de software funcional y validado en la nube.

La estructura del Sprint Backlog detallada en la Tabla LXXV pone de manifiesto la línea lógica de construcción de la arquitectura Serverless; empezando por la infraestructura base (Frontend e Identidad) y siguiendo hacia el engine de evaluación gamificada (backend algorítmico), hasta alcanzar finalmente los módulos de analítica y validación empírica en producción.

Tabla LXXV: Sprint Backlog detallado por iteración (Septiembre 2025 – Marzo 2026)

Sprint	Periodo	Tareas Técnicas y Componentes AWS	HU	Estado
S1	Septiembre 2025	**Infraestructura Base:** Configurar *bucket* Amazon S3 como sitio estático; creación de rutas base; configuración de Amazon Cognito (User Pool) y despliegue inicial de API Gateway.	HU-01	Hecho
S2	Octubre 2025	**Autenticación y Auditoría:** Programación del flujo PKCE en *frontend*; despliegue de Lambda `saveLoginLog`; persistencia en tabla `logins` de DynamoDB.	HU-01	Hecho
S3	Noviembre 2025	**Motor de Examen V1:** Despliegue de Lambda `obtenerPreguntas`; UI de opción múltiple con validación; *feedback* inmediato en el DOM (*Heads-Up Display*).	HU-02 HU-06	Hecho

Continúa en la siguiente página

Tabla LXXV (continuación). Sprint Backlog detallado por iteración

Sprint	Periodo	Tareas Técnicas y Componentes AWS	HU	Estado
S4	Diciembre 2025	**Lógica de Gamificación Central:** Implementación del control de intentos, desbloqueo progresivo de niveles y persistencia temporal del estado de la evaluación.	HU-05	Hecho
S5	Enero 2026	**Reactivos de Completación:** Desarrollo del motor *frontend* y calificador en Lambda (uso de Expresiones Regulares para normalizar tildes/espacios).	HU-03	Hecho
S6	Febrero 2026	**Reactivos de Emparejamiento:** Lógica *Drag & Drop* para ejercicios de *matching*; calificador proporcional de aciertos en backend.	HU-04	Hecho
S7	Febrero 2026	**Clasificación Global (*Leaderboard*):** Creación de Índice Secundario Global (GSI) en DynamoDB; Lambda `getLeaderboard` para asignar medallas Top 3.	HU-07 HU-08	Hecho
S8	Inicios de Marzo 2026	**Analítica y Progreso:** Despliegue de Lambda `getMetrics`; integración con *Chart.js* para visualizar % de avance; módulo de carga de reactivos.	HU-09	Hecho

Continúa en la siguiente página

Tabla LXXV (continuación). Sprint Backlog detallado por iteración

Sprint	Periodo	Tareas Técnicas y Componentes AWS	HU	Estado
S9	Finales de Marzo 2026	**Pruebas en Producción (Piloto):** Ejecución con N=50 usuarios; resolución de concurrencia en Amazon Cognito para estabilizar accesos simultáneos; recolección de métricas SUS.	HU-10	Hecho

Elaboración propia.

3.17.9 Planificación de Tareas y Estimación de Esfuerzo (Sprints 1 al 9)

Para evidenciar el trabajo de ingeniería a nivel de código y justificar el cronograma del proyecto, se desglosan a continuación las tareas técnicas correspondientes a las nueve iteraciones del ciclo de vida, cuantificando el esfuerzo en horas invertidas.

El primer Sprint estuvo centrado en crear la infraestructura básica del proyecto en las nubes. En esta fase, se aprovisionaron los servicios básicos para alojamiento web, gestión de identidad y configuración del ruteo del interfaz de programa aplicación (API).

Tabla LXXVI: Detalle de Tareas: Sprint 1 (Infraestructura Base)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T1.1	Configuración del bucket de Amazon S3 para el alojamiento estático del frontend.	4
T1.2	Aprovisionamiento de User Pool en Amazon Cognito y configuración de dominios seguros.	5

T1.3	Despliegue de API Gateway y habilitación de recursos Proxy.	4
T1.4	Configuración de variables de entorno e inicialización del archivo config.js.	2
Total		**15h**

Una vez consolidada la infraestructura, el segundo sprint se dedicó a la implementación de la capa de seguridad y auditoría. Se desarrolló el flujo de autenticación y se configuró el registro automático de accesos para garantizar la trazabilidad de los usuarios en la base de datos.

Tabla LXXVII: Detalle de Tareas: Sprint 2 (Autenticación y Auditoría)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T2.1	Implementación de utilidades criptográficas PKCE (verifier y challenge) en auth.js.	5
T2.2	Configuración del intercambio de tokens JWT con Amazon Cognito.	4
T2.3	Desarrollo de la función Lambda saveLoginLog.mjs para captura de UserAgent e IP.	4
T2.4	Verificación de persistencia de logs en la tabla NoSQL de DynamoDB.	2
Total		**15h**

El tercer sprint abordó la construcción del motor principal de evaluación para preguntas de opción múltiple. Las tareas incluyeron la programación de la lógica de aleatoriedad, la gestión del tiempo mediante un temporizador regresivo y la presentación dinámica de las preguntas en la interfaz.

Tabla LXXVIII: Detalle de Tareas: Sprint 3 (Motor de Evaluación MCQ)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T3.1	Programación de obtenerPreguntas.mjs integrando el algoritmo de barajado aleatorio.	4
T3.2	Desarrollo de la interfaz prueba.html con renderizado dinámico de opciones.	5
T3.3	Implementación del temporizador regresivo global (600 segundos) en juegos.js.	3
T3.4	Renderizado de retroalimentación inmediata visualizando la explicación académica.	3
Total		**15h**

Durante el cuarto sprint, se implementaron las lógicas centrales de gamificación y control de progresión. Se establecieron las reglas de negocio para limitar los intentos de evaluación, así como las restricciones para el desbloqueo progresivo de los niveles de mayor dificultad.

Tabla LXXIX: Detalle de Tareas: Sprint 4 (Lógica de Gamificación Central)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T4.1	Programación de la lógica de control de intentos (máximo 3) en estado.js.	5
T4.2	Implementación algorítmica para renovación de intentos tras 60 minutos.	4
T4.3	Desarrollo de barreras de acceso (desbloqueo de Normal y Difícil) en seleccion.js.	4

T4.4	Persistencia temporal en almacenamiento local y limpieza dinámica post examen.	2
Total		**15h**

El quinto sprint extendió las capacidades del motor de evaluación para soportar preguntas de completación. Se programaron algoritmos de normalización de texto para validar las respuestas escritas por el usuario, omitiendo variaciones ortográficas como mayúsculas y tildes.

Tabla LXXX: Detalle de Tareas: Sprint 5 (Motor de Completación)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T5.1	Implementación de la variante de texto libre en el motor de renderizado de juegos.js.	4
T5.2	Normalización de cadenas de texto empleando expresiones regulares en el cliente.	5
T5.3	Adaptación del bloque de datos de respuestas para consolidación y envío a AWS Lambda.	3
T5.4	Ajustes en la interfaz gráfica para validación de ingresos textuales.	2
Total		**14h**

En el sexto sprint se desarrolló el módulo para los reactivos de emparejamiento. Esta iteración requirió la implementación de interacciones dinámicas en el lado del cliente y algoritmos de calificación proporcional en el servidor para evaluar aciertos parciales.

Tabla LXXXI: Detalle de Tareas: Sprint 6 (Reactivos de Emparejamiento)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T6.1	Modificación de obtenerPreguntas.mjs para procesar el atributo de parejas.	4
T6.2	Implementación de lógica interactiva de selección encadenada en juegos.js.	6
T6.3	Desarrollo del algoritmo de cálculo de aciertos parciales para asignación de puntaje.	3
T6.4	Pruebas unitarias de iteración y calificación proporcional.	2
Total		**15h**

El séptimo sprint se centró en el componente competitivo de la gamificación. Se construyó el sistema de clasificación global mediante índices secundarios en la base de datos y se programó la lógica visual para la asignación de medallas a los puntajes más altos.

Tabla LXXXII: Detalle de Tareas: Sprint 7 (Gamificación y Ranking)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T7.1	Configuración del índice secundario global en la tabla de resultados de DynamoDB.	3
T7.2	Programación del microservicio getLeaderboard.mjs para consultas descendentes.	5
T7.3	Desarrollo de rankings.js para inserción de componentes visuales representativos de medallas.	4

T7.4	Pruebas de renderizado para resaltar dinámicamente la fila del usuario con sesión activa.	2
Total		**14h**

El octavo sprint estuvo orientado a proveer herramientas de inteligencia de datos para el rol administrador. Se consolidó el panel analítico con gráficos interactivos y se habilitó la carga masiva de bancos de preguntas para facilitar la gestión del contenido.

Tabla LXXXIII: Detalle de Tareas: Sprint 8 (Analítica Administrativa)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T8.1	Lógica de agregación matricial en getMetrics.mjs para calculo de errores y embudos.	5
T8.2	Integración de la librería Chart.js en admin.js para visualización interactiva.	4
T8.3	Desarrollo del script de carga masiva addQuestion.mjs y validación en cliente.	4
T8.4	Filtro de seguridad para autorizar consultas solo al grupo con privilegios administrativos.	2
Total		**15h**

El último sprint cerró el ciclo de desarrollo con el despliegue del sistema en un entorno de producción y la ejecución de la prueba piloto. Se recolectaron los datos de la escala de usabilidad y se automatizó el cálculo de las métricas de fiabilidad estadística.

Tabla LXXXIV: Detalle de Tareas: Sprint 9 (Piloto en Producción y Validación)

ID Tarea	Actividad Técnica (Referencia a Código)	Horas
T9.1	Ejecución del piloto controlado con muestra de estudiantes en infraestructura AWS.	6
T9.2	Extracción de métricas crudas desde DynamoDB y exportación a formato de tabla.	3
T9.3	Desarrollo de script automatizado en Python para calcular Varianza y Alfa de Cronbach.	4
T9.4	Recopilación de datos de la escala SUS y cálculo matemático de usabilidad.	3
Total		**16h**

3.17.10 Métricas de Control: Velocidad y Gráfico de Burndown

En la gestión ágil, la capacidad de entrega se mide a través de la *Velocidad*, definida como el número de puntos de historia (Story Points) que el equipo completa satisfactoriamente en cada iteración [44]. Esta métrica permitió estabilizar el ritmo de desarrollo y predecir con mayor exactitud la fecha de despliegue del piloto final.

Tabla LXXXV: Velocidad de desarrollo por Sprint (Puntos Planificados vs. Completados)

Sprint	Pts Planificados	Pts Completados	Observaciones Técnicas
S1	5	5	Despliegue exitoso de infraestructura base S3/Cognito.
S2	5	5	Estabilización de la auditoría en DynamoDB.

S3	5	5	Motor de opción múltiple validado.
S4	3	3	Control de intentos y desbloqueo progresivo de niveles.
S5	5	5	Finalización del motor de completación con Regex.
S6	5	5	Implementación de *Drag & Drop* para *matching*.
S7	3	3	Configuración de GSI y Ranking global.
S8	4	3	Integración del panel analítico y métricas administrativas.
S9	5	5	Aplicación de validación SUS con 50 usuarios.

Análisis del Burndown Chart (Sprint 8) Para monitorear el progreso diario y detectar cuellos de botella, se utilizó el gráfico de *Burndown*. En la Tabla LXXXVI se detalla el comportamiento del Sprint 8, el cual representó el hito de mayor carga visual y analítica del proyecto.

Tabla LXXXVI: Registro de Burndown del Sprint 8 (Puntos restantes por día)

Día	**Ideal (pts)**	**Actual (pts)**	**Observación de la Iteración**
Lunes	4.0	4.0	Inicio de la integración de *Chart.js*.
Martes	3.5	3.5	Calificador de insignias operando en Lambda.
Miércoles	3.0	3.2	Ajustes generan desfase del DOM del mapa visual.
Jueves	2.5	2.5	Las consultas sincronizan GSI para medallas.

Viernes	2.0	2.2	Las métricas de error se integran en el dashboard.
Sábado	1.0	1.2	Pruebas unitarias de carga de archivos JSON.
Domingo	0.0	1.0	Pendiente: Validación final del panel analítico durante el piloto en producción.

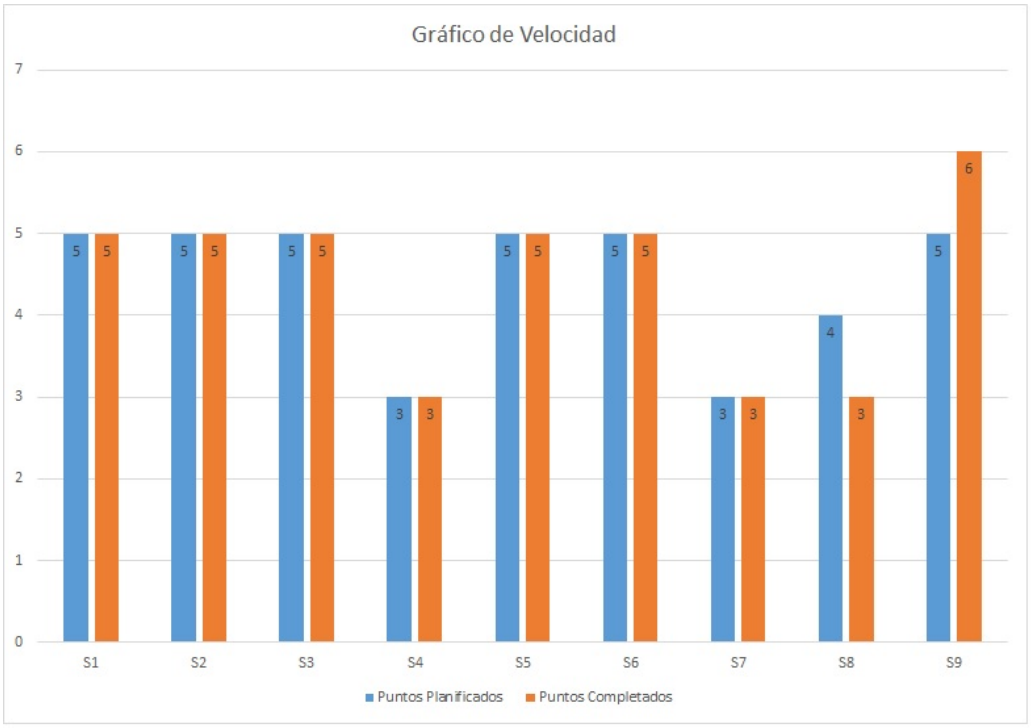


Figura 16.
Gráfico de velocidad: Puntos planificados vs. completados por cada Sprint.

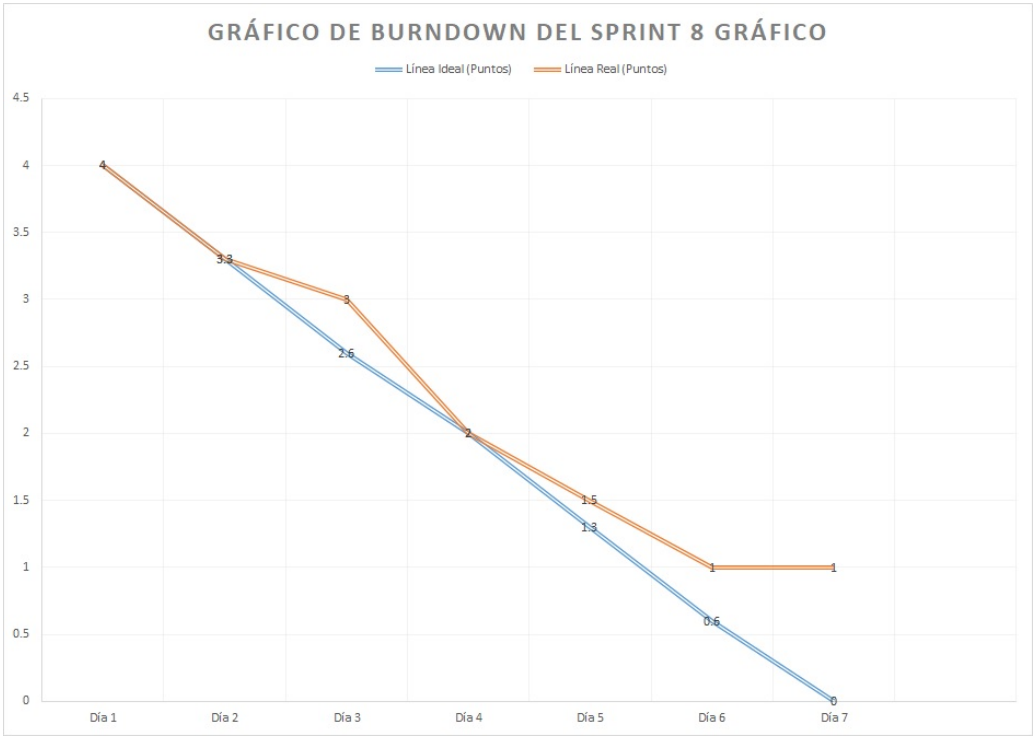


Figura 17.
Gráfico de Burndown del Sprint 8: Seguimiento diario del trabajo pendiente.

3.18 Trazabilidad del Diseño de Gamificación (MDA)

Para asegurar que la arquitectura *Serverless* cumpla con el propósito pedagógico de motivar al estudiante, se aplicó la matriz de trazabilidad del modelo MDA.

Tabla LXXXVII: Matriz de trazabilidad MDA aplicada a la plataforma

Mecánica (Código/AWS)	**Dinámica (Comportamiento)**	**Estética (Emoción)**
Puntuación algorítmica y precisión porcentual en Lambda.	Evaluación continua con feedback inmediato sobre el desempeño.	Sentimiento de desafío, conciencia de la competencia intelectual.
Insignias de nivel y medallas Top 3 vía GSI de DynamoDB.	El reconocimiento de la categoría pública del valor de su esfuerzo.	Orgullo, estatus social y reconocimiento entre pares.

Desbloqueo condicionado de niveles (Normal/Difícil).	Progresión por hitos - El contenido solo está disponible si dominas previamente.	Sensación de satisfacción al dominar temas y sentimiento de progreso.
Cronómetro de 600 segundos y mapa de navegación visual.	Administración de tiempos con presiones y autogestión al progreso del examen.	La ansiólisis por desafío: reducción de la ansiedad vía control ambiental.

Delimitación de la Validación del Modelo MDA: Cabe recalcar que en la presente investigación, el marco MDA se empleó estrictamente como guía metodológica para el diseño de la arquitectura de *software* orientada a la motivación. La validación empírica del impacto emocional (las Estéticas) se delimitó a un análisis cuantitativo utilizando métricas de uso como *proxy* de *engagement*. En lugar de aplicar instrumentos psicométricos adicionales, se utilizaron los datos de retención, intentos repetidos y uso del *Leaderboard* capturados automáticamente por el módulo analítico (CU-07) y persistidos en Amazon DynamoDB como evidencia empírica directa de la motivación generada, dejando la evaluación psicológica fuera del alcance estricto de este proyecto de ingeniería de *software*.

3.18.1 Trazabilidad Pedagógica

Tabla LXXXVIII: Mapeo de objetivos de aprendizaje con mecánicas de gamificación

Objetivo de Aprendizaje	**Mecánicas Asociadas**	**Indicador de Éxito**
Mejorar el dominio de contenidos preuniversitarios.	Retroalimentación formativa y reactivos multiformato.	Incremento en el porcentaje de precisión por categoría.
Se sostiene la motivación intrínseca en el estudio.	Medallas Top 3 y etiquetas de nivel (*Rookie* a *Master*).	Frecuencia de uso y posición en el *leaderboard*.

Se fomenta la autorregulación del tiempo de examen.	HUD dinámico y temporizadores regresivos por Lambda.	Se reduce el tiempo promedio de resolución por intento.

3.19 Metodología de Validación: Escala de Usabilidad del Sistema (SUS)

La evaluación de la plataforma *PreServerless* incorpora la medición de la percepción subjetiva de usabilidad mediante la Escala de Usabilidad del Sistema (SUS), un instrumento reconocido por ser una herramienta robusta, de bajo costo y alta fiabilidad para mediciones globales en entornos tecnológicos [10]. Este método permite obtener una visión general de las valoraciones subjetivas tras la interacción con el sistema, centrándose en dimensiones de efectividad, eficiencia y satisfacción [10].

3.19.1 Fundamentación del Instrumento

Se aplicó la versión completa de 10 ítems validada para la población hispanohablante por Castilla et al. (2023), la cual mantiene una consistencia interna sólida con un alfa de Cronbach de 0.76 [51]. La escala se basa en un diseño de tipo Likert donde el encuestado indica su grado de acuerdo o desacuerdo en un rango de 1 a 5 [10]. El instrumento alterna sistemáticamente afirmaciones positivas y negativas para mitigar sesgos de respuesta y asegurar que el evaluador procese críticamente cada reactivo [10], [51].

A continuación, se presentan en la **Tabla LXXXIX** los reactivos literales utilizados durante la fase de validación, los cuales mantienen la neutralidad técnica requerida para la evaluación.

Tabla LXXXIX: Ítems de la escala SUS aplicados a la plataforma PreServerless

No.	Afirmación del Instrumento	Polaridad
1	Creo que me gustaría usar este sistema frecuentemente.	Positiva
2	Encontré el sistema innecesariamente complejo.	Negativa

Continúa en la siguiente página

Tabla LXXXIX: Tabla LXXXIX (Continuación)

No.	Afirmación del Instrumento	Polaridad
3	Pensé que el sistema era fácil de usar.	Positiva
4	Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema.	Negativa
5	Encontré que las diversas funciones de este sistema estaban bien integradas.	Positiva
6	Pensé que había demasiada inconsistencia en este sistema.	Negativa
7	Imagino que la mayoría de las personas podrían aprender a usar este sistema muy rápidamente.	Positiva
8	Encontré el sistema muy engorroso de usar.	Negativa
9	Me sentí muy confiado usando el sistema.	Positiva
10	Necesité aprender muchas cosas antes de poder comenzar a usar este sistema.	Negativa

Continúa en la siguiente página

Nota: Reactivos obtenidos de la validación española de Castilla et al. (2023).

3.19.2 Protocolo de Aplicación con Usuarios Reales

La validación se ejecutó mediante una prueba piloto con una muestra de 50 estudiantes. El procedimiento se estructuró en las siguientes fases:

1. **Interacción Directa:** El estudiante interactuó con la plataforma completando un flujo de usua-

rio real: autenticación, selección de simulacro y resolución de una prueba con retroalimentación inmediata.

2. **Administración del Cuestionario:** Inmediatamente después de la interacción, el usuario registró su respuesta subjetiva a cada ítem en una escala de 1 (”Totalmente en desacuerdo”) a 5 (”Totalmente de acuerdo”) [10].

3. **Saturación de Datos:** Se instruyó a los usuarios a marcar el punto central de la escala en caso de duda para garantizar que todos los ítems fueran completados [10].

3.19.3 Algoritmo de Calificación y Escalamiento

Para obtener una medida compuesta de la usabilidad, los puntajes individuales se transforman siguiendo las reglas de Brooke [10]:

- **Ítems impares (1, 3, 5, 7, 9):** La contribución al puntaje es la posición de la escala menos 1 ($X_i - 1$) [10], [51].
- **Ítems pares (2, 4, 6, 8, 10):** La contribución es 5 menos la posición de la escala marcada ($5 - X_i$) [10], [51].
- **Puntuación Final:** Se suma el aporte de los 10 ítems y se multiplica por 2.5 para obtener un valor final en el rango de 0 a 100 [10], [51].

3.19.4 Criterios de Interpretación de Resultados

Los resultados se interpretan mediante los rangos de aceptabilidad y descriptores adjetivos de Bangor et al. (2008), detallados en la **Tabla XC** [51], [52].

Tabla XC: Rangos de interpretación para la puntuación SUS

Puntaje (Score)	**Calificación Adjetiva**	**Aceptabilidad**
$> 80{,}3$	Excelente (*Excellent*)	Aceptable
$68{,}0 - 80{,}3$	Bueno (*Good*)	Aceptable
$51{,}0 - 67{,}9$	Regular (*OK*)	Marginal
$< 51{,}0$	Pobre / Deficiente	No Aceptable

Fuente: Basado en Bangor et al. (2008) y Castilla et al. (2023).

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

En este capítulo se presentan y analizan los datos obtenidos tras la implementación de la plataforma educativa *PreServerless*. El análisis se divide en tres fases: la validación estadística de la muestra (normalidad), la evaluación de usabilidad bajo el estándar ISO/IEC 25010 utilizando la escala SUS, y la cuantificación de los impactos del proyecto.

4.1 Análisis de Fiabilidad y Normalidad

Antes de proceder con la interpretación descriptiva, es metodológicamente indispensable determinar la distribución estadística de los datos recolectados de los 50 usuarios. Este análisis define si la muestra sigue un comportamiento paramétrico (curva simétrica) o no paramétrico, lo cual dictará las medidas de tendencia central correctas a utilizar.

4.1.1 Prueba de Normalidad (Kolmogorov-Smirnov)

Para el presente proyecto, se cuenta con una muestra efectiva de $n = 50$ usuarios. De acuerdo con la literatura estadística, para muestras de tamaño $n \geq 50$, la prueba de normalidad adecuada y de mayor potencia es el estadístico de **Kolmogorov-Smirnov (K-S)** con la corrección de Lilliefors.

Esta prueba, teóricamente, contrasta el valor observado de la función de distribución acumulada empírica, con una distribución normal teórica. Un estadístico D evaluado representa la distancia absoluta máxima entre las dos curvas. Si esta divergencia resulta en un valor p inferior a 0.05, se puede concluir que los datos no vienen de una distribución normal.

En la **Tabla XCI** se presentan los resultados procesados algorítmicamente para el Puntaje Global SUS obtenido por los participantes.

Tabla XCI: Prueba de Normalidad Kolmogorov-Smirnov para Escala SUS

	Kolmogorov-Smirnov		
Variable de Evaluación	**Estadístico (D)**	**gl**	**Sig. (p-valor)**
Puntaje Global SUS	0.169	50	0.001

Para corroborar gráficamente este comportamiento matemático, se generó un histograma de densidad (**Figura 18**) que superpone las calificaciones reales de los usuarios frente a la silueta de la campana de Gauss ideal.

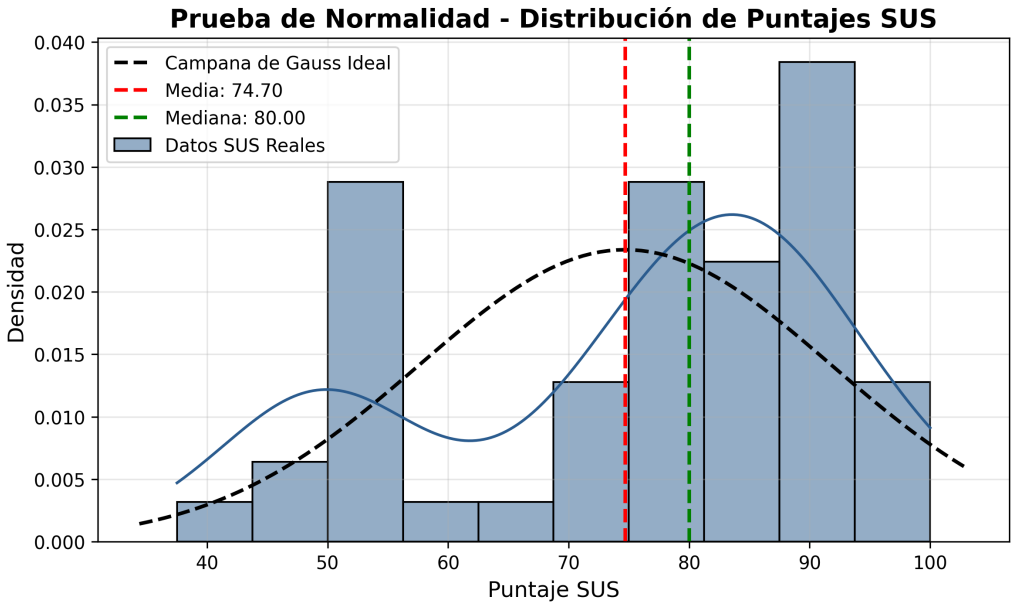


Figura 18.
Distribución empírica del Puntaje Global SUS frente a la curva normal teórica.

Interpretación de la Prueba: Como puede verse en la gráfica y como puede confirmarse en la tabla, el nivel de significancia es p = 0.001. Como es p <0.05 rechazamos la hipótesis nula de normalidad perfecta. A nivel visual, esto se traduce como un sesgo negativo evidente: los datos se amontonan a la derecha de la escala. Del punto de vista del ingeniero del software, este descubrimiento es alentador: La mayoría de usuarios congregaron sus respuestas alrededor de los rangos de excelencia. En consecuencia, se empleará al median (80.00) en cuanto estimador de la Usabilidad Principal.

4.2 Análisis e Interpretación de Resultados de Usabilidad

4.2.1 Población y Muestra

El universo total identificado comprende $N = 55$ individuos. Para comprobar la representatividad, se aplicó la fórmula para poblaciones finitas:

$$n = \frac{Z^2 \cdot p \cdot q \cdot N}{e^2 \cdot (N-1) + Z^2 \cdot p \cdot q} \tag{4.1}$$

Considerando un nivel de confianza del 95 % ($Z = 1{,}96$), una probabilidad de éxito del 50 % ($p = 0{,}5$) y un margen de error del 5 % ($e = 0{,}05$):

$$n = \frac{(1{,}96)^2 \cdot 0{,}5 \cdot 0{,}5 \cdot 55}{(0{,}05)^2 \cdot (54) + (1{,}96)^2 \cdot 0{,}5 \cdot 0{,}5} \approx 48{,}22 \tag{4.2}$$

El tamaño mínimo requerido es de 49 estudiantes. Se obtuvo una muestra efectiva de **50 usuarios**, lo que otorga total validez estadística al estudio.

4.2.2 Tabulación de Frecuencias de la Escala SUS

En la **Tabla XCII** se consolida la matriz de frecuencias para los 10 reactivos aplicados a los 50 usuarios.

Tabla XCII: Distribución de frecuencias de la escala SUS ($n = 50$).

Reactivo (Pregunta)	**1**	**2**	**3**	**4**	**5**
1. Creo que me gustaría usar este sistema con frecuencia.	0	2	4	19	25

Tabla XCII: 569XCII (Continuación)

Reactivo (Pregunta)	**1**	**2**	**3**	**4**	**5**
2. Encontré el sistema innecesariamente complejo.	12	22	9	2	5
3. Pensé que el sistema era fácil de usar.	0	1	7	15	27
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.	20	10	10	4	6
5. Encontré que las diversas funciones de este sistema estaban bien integradas.	0	1	8	19	22
6. Pensé que había demasiada inconsistencia en este sistema.	17	16	11	2	4
7. Imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.	0	1	8	12	29
8. Encontré el sistema muy engorroso de usar.	17	14	10	5	4
9. Me sentí muy seguro usando el sistema.	0	2	7	14	27
10. Necesité aprender muchas cosas antes de poder seguir con este sistema.	18	9	7	6	10

Nota: 1=Totalmente en desacuerdo, 2=En desacuerdo, 3=Neutral, 4=De acuerdo, 5=Totalmente de acuerdo.

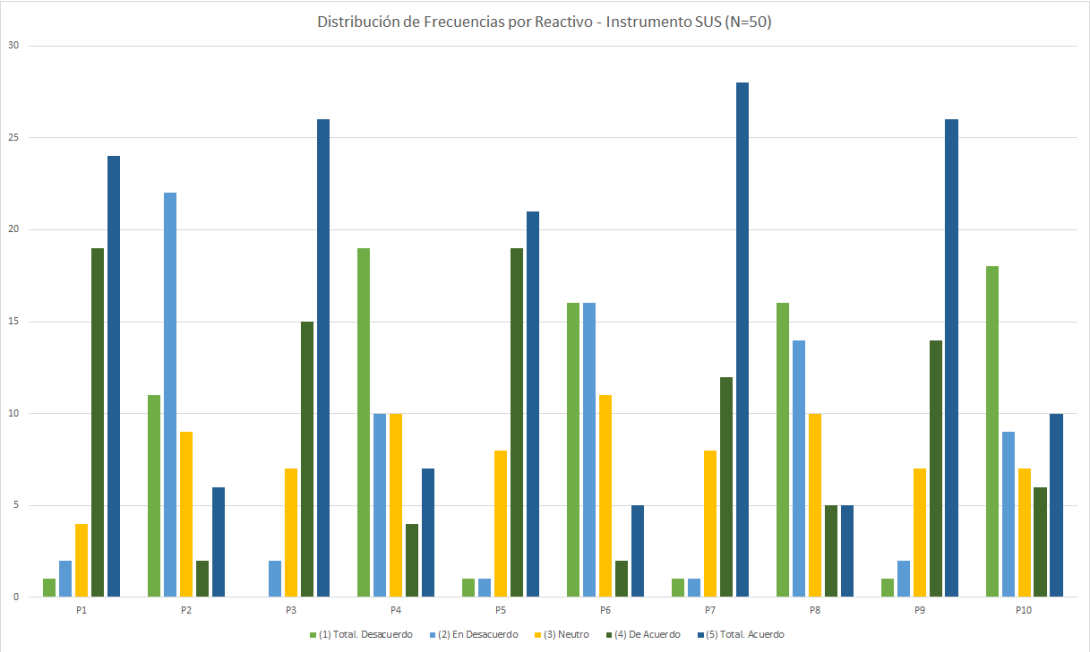


Figura 19.
Distribución de Frecuencias por Reactivo - Instrumento SUS (N=50)

4.2.3 Análisis Descriptivo por Preguntas

A continuación se detalla el análisis de las tres primeras interrogantes aplicadas a los 50 usuarios del sistema.

Pregunta 1: Creo que me gustaría usar este sistema con frecuencia

Tabla XCIII: Frecuencias y porcentajes de la Pregunta 1

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	0	0 %
2. En desacuerdo	2	4 %
3. Neutral	4	8 %
4. De acuerdo	19	38 %

Escala de Likert	Frecuencia (f)	Porcentaje (%)
5. Totalmente de acuerdo	25	50 %
Total	**50**	**100 %**

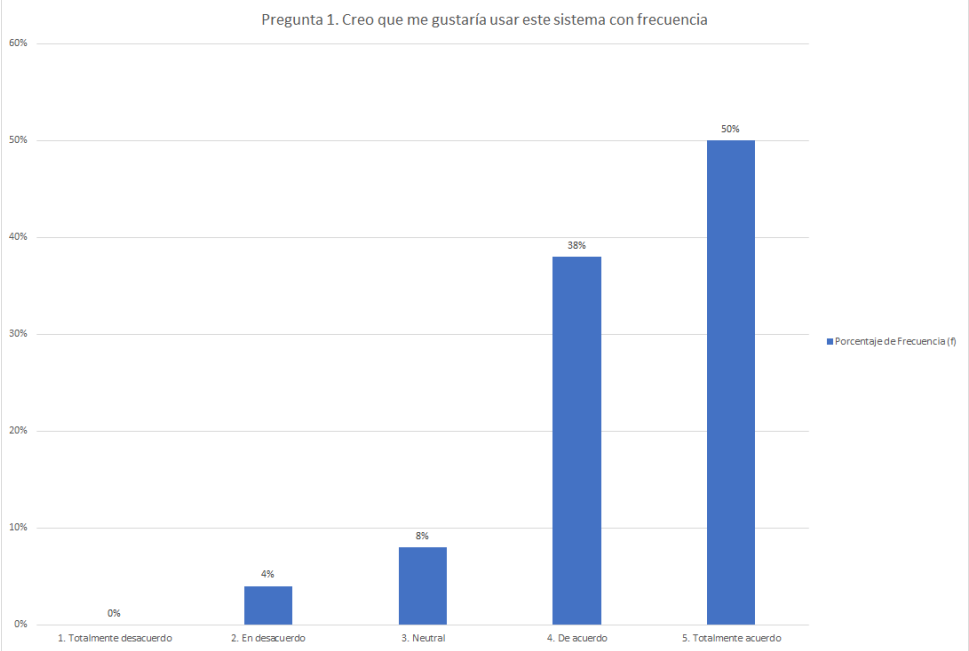


Figura 20.
Distribución de respuestas para la Pregunta 1.

Análisis: En la Pregunta 1 el 88 % de los usuarios (44 personas) manifiesta una tendencia positiva al uso habitual del sistema. Estudio constante atrae; este alto nivel de aceptación indica una interesante potencialidad con la plataforma PreServerless.

Pregunta 2: Encontré el sistema innecesariamente complejo

Tabla XCIV: Frecuencias y porcentajes de la Pregunta 2

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	11	22 %
2. En desacuerdo	22	44 %
3. Neutral	9	18 %
4. De acuerdo	2	4 %
5. Totalmente de acuerdo	6	12 %
Total	**50**	**100 %**

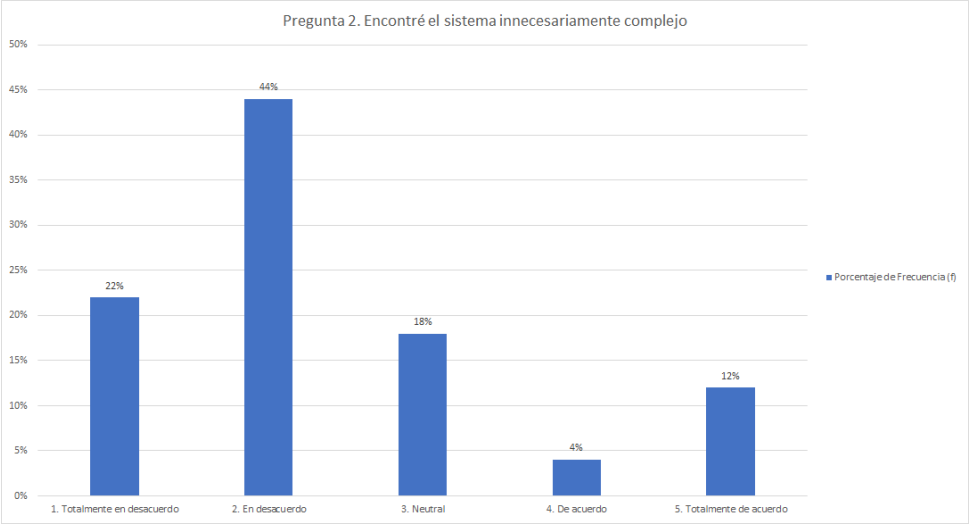


Figura 21.
Distribución de respuestas para la Pregunta 2.

Análisis: Como es un tipo de pregunta invertida, el 66 % rechazó la idea de que el sistema es muy complejo. Eso prueba que con esto se ha simplificado al usuario final procesos técnicos del sistema.

Pregunta 3: Pensé que el sistema era fácil de usar

Tabla XCV: Frecuencias y porcentajes de la Pregunta 3

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	0	0 %
2. En desacuerdo	1	2 %
3. Neutral	7	14 %
4. De acuerdo	15	30 %
5. Totalmente de acuerdo	27	54 %
Total	**50**	**100 %**

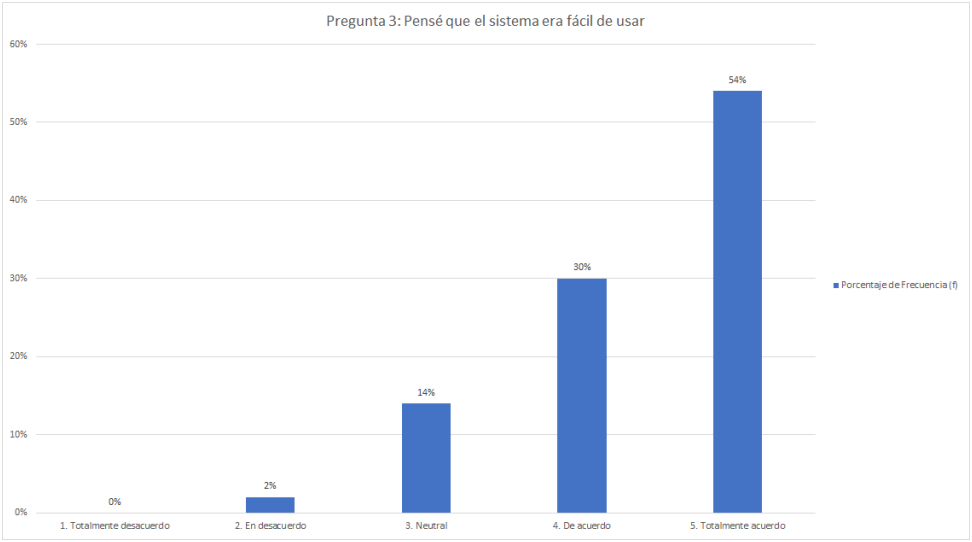


Figura 22.
Distribución de respuestas para la Pregunta 3.

Análisis: Un considerable 84 % de los encuestados piensan que es un sistema muy fácil de usar. La falta de respuesta en ”Totalmente en Desacuerdoratifica el poder de la arquitectura de front end con respecto a la usabilidad.

Pregunta 4: Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema

Tabla XCVI: Frecuencias y porcentajes de la Pregunta 4

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	20	40 %
2. En desacuerdo	10	20 %
3. Neutral	10	20 %
4. De acuerdo	4	8 %
5. Totalmente de acuerdo	6	12 %
Total	**50**	**100 %**

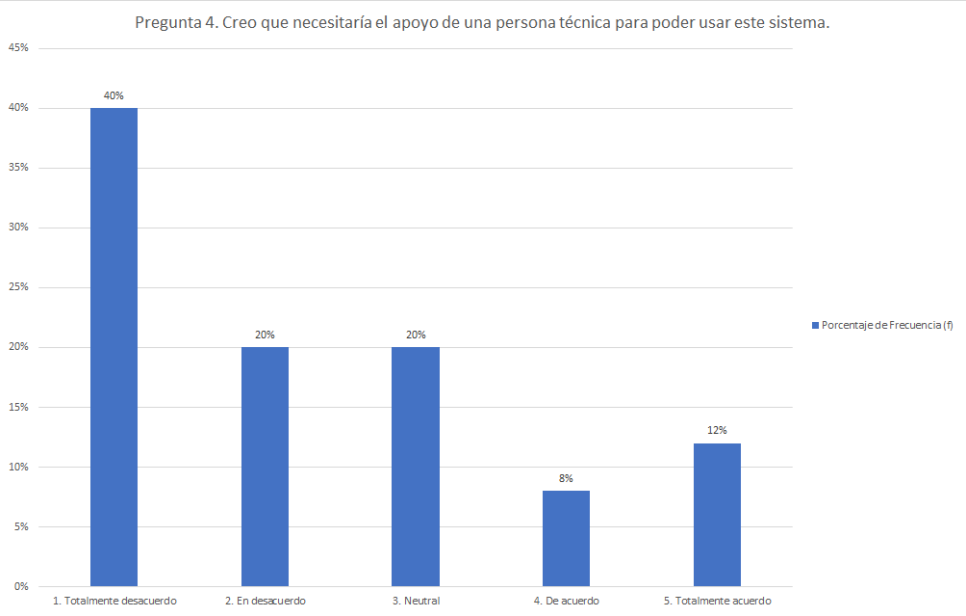


Figura 23.
Distribución de respuestas para la Pregunta 4.

Análisis: Al ser una pregunta de carácter invertido, el 60 % de los usuarios (30 personas) manifiesta no requerir asistencia técnica para operar la plataforma. Este resultado valida la autonomía que brinda el sistema al estudiante, confirmando que la interfaz es lo suficientemente intuitiva para su uso independiente.

Pregunta 5: Encontré que las diversas funciones de este sistema estaban bien integradas

Tabla XCVII: Frecuencias y porcentajes de la Pregunta 5

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	0	0 %
2. En desacuerdo	1	2 %
3. Neutral	8	16 %
4. De acuerdo	19	38 %
5. Totalmente de acuerdo	22	44 %
Total	**50**	**100 %**

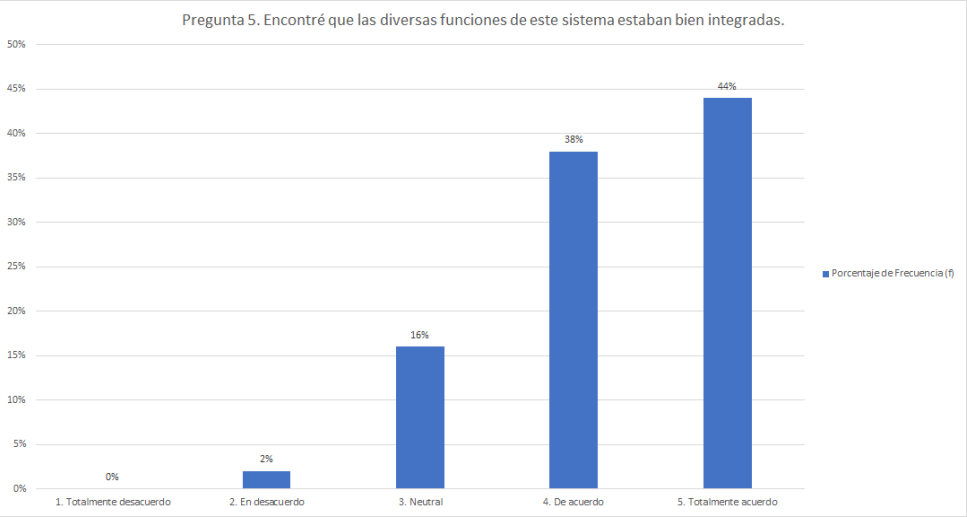


Figura 24.
Distribución de respuestas para la Pregunta 5.

Análisis: El 82 % de la gente coincide en que las características del sistema están muy bien integradas. Aunque la arquitectura sin servidor utilice microservicios desacoplados, el usuario percibe al sistema como algo cohesionado, lo cual subraya una correcta orquestación de los servicios de AWS.

Pregunta 6: Pensé que había demasiada inconsistencia en este sistema

Tabla XCVIII: Frecuencias y porcentajes de la Pregunta 6

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	17	34 %
2. En desacuerdo	16	32 %
3. Neutral	11	22 %
4. De acuerdo	2	4 %
5. Totalmente de acuerdo	4	8 %
Total	**50**	**100 %**

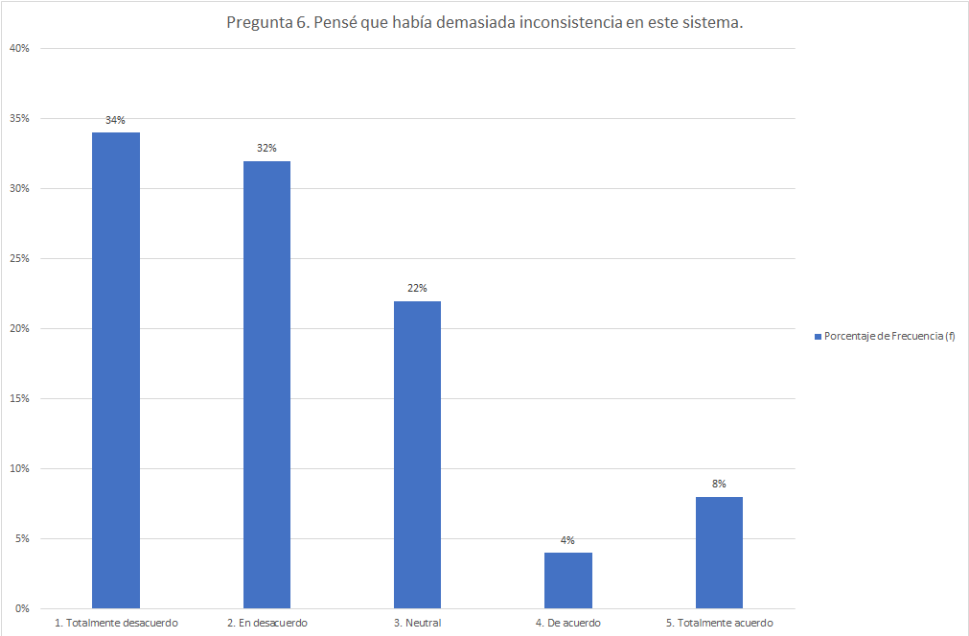


Figura 25.
Distribución de respuestas para la Pregunta 6.

Análisis: En esta pregunta invertida (¿existen inconsistencias en el sistema?), el 66 % de la muestra niega su existencia. Una baja percepción de errores o comportamientos inesperados refrenda la robustez del Lógica de Negocio aplicada a la función Lambda y su estabilidad frente la concurrencia de usuario.

Pregunta 7: Imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente

Tabla XCIX: Frecuencias y porcentajes de la Pregunta 7

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	0	0 %
2. En desacuerdo	1	2 %
3. Neutral	8	16 %
4. De acuerdo	12	24 %
5. Totalmente de acuerdo	29	58 %
Total	**50**	**100 %**

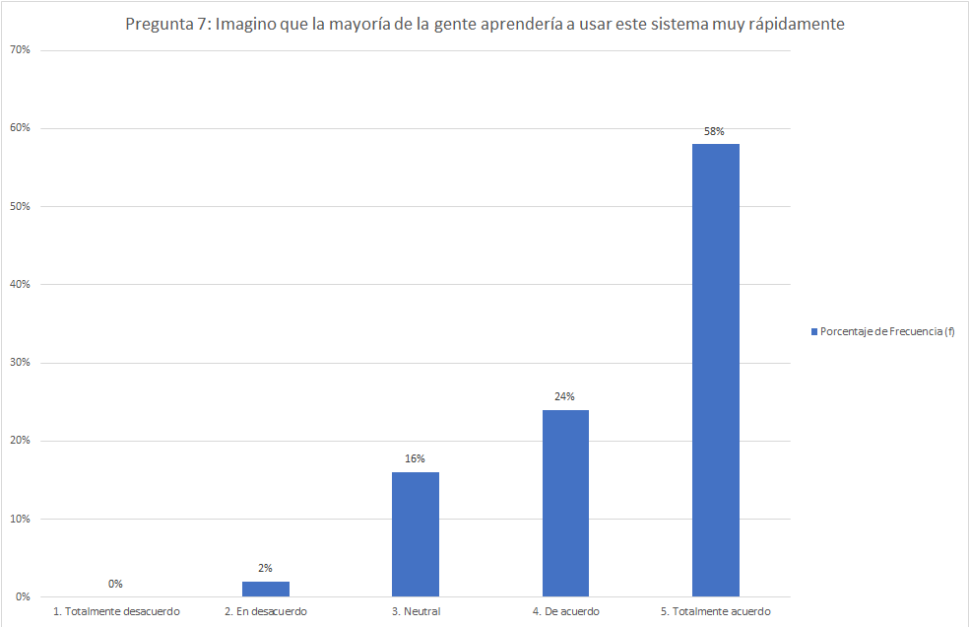


Figura 26.
Distribución de respuestas para la Pregunta 7.

Análisis: Con una tajante mayoría del 82 % (41 personas), expresa que el sistema tiene un rápido aprendizaje. El nivel cero de respuestas a los niveles de desagrado demuestra el mínimo de curva de

aprendizaje de esta plataforma, que resulta ser un factor crítico de éxito en preparación preuniversitaria autoguiada.

Pregunta 8: Encontré el sistema muy engorroso de usar

Tabla C: Frecuencias y porcentajes de la Pregunta 8

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	17	34 %
2. En desacuerdo	14	28 %
3. Neutral	10	20 %
4. De acuerdo	5	10 %
5. Totalmente de acuerdo	4	8 %
Total	**50**	**100 %**

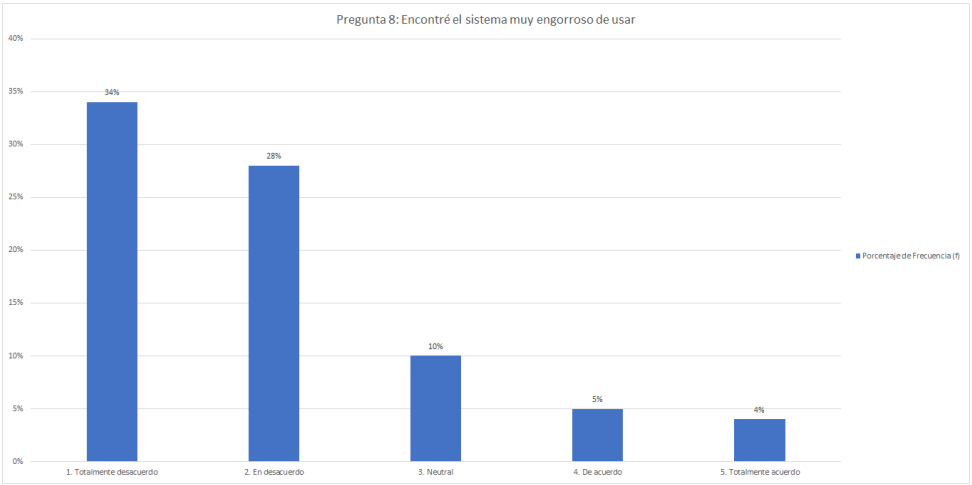


Figura 27.
Distribución de respuestas para la Pregunta 8.

Análisis: Por ser un problema con naturaleza invertida, el 62 % de las personas rechazan la percepción en cuanto al sistema como entorpecedor. Este es un buen resultado que refleja los beneficios que trae

consigo la arquitectura Serverless, en donde la baja latencia y el proceso eficiente de datos ayudan a crear agilidad operacionalmente.

Pregunta 9: Me sentí muy seguro usando el sistema

Tabla CI: Frecuencias y porcentajes de la Pregunta 9

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	0	0 %
2. En desacuerdo	2	4 %
3. Neutral	7	14 %
4. De acuerdo	14	28 %
5. Totalmente de acuerdo	27	54 %
Total	**50**	**100 %**

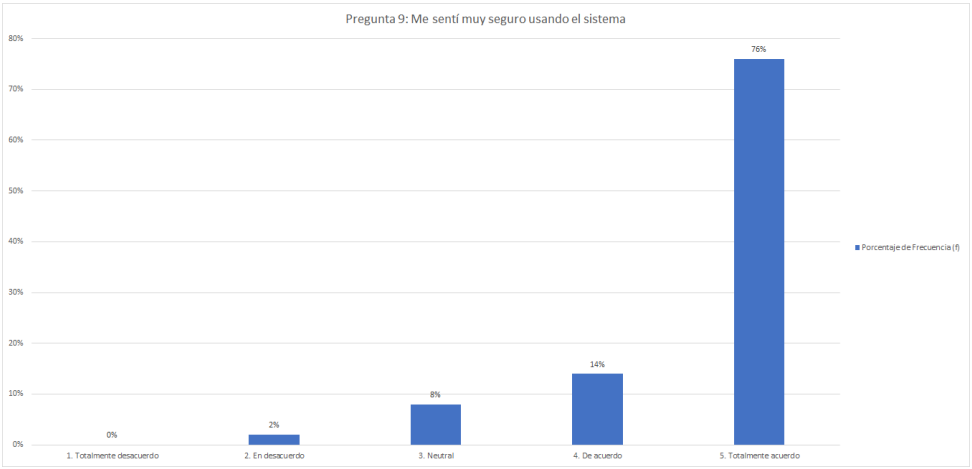


Figura 28.
Distribución de respuestas para la Pregunta 9.

Análisis: La confianza del usuario es el ítem con mayor calificación positiva, donde el 82 % (41 usuarios) se siente *Totalmente de acuerdo* con la seguridad del sistema. El uso de servicios gestionados

de AWS para el manejo de sesiones y datos garantiza una percepción de estabilidad y robustez tecnológica que se refleja en este indicador.

Pregunta 10: Necesité aprender muchas cosas antes de poder seguir con este sistema

Tabla CII: Frecuencias y porcentajes de la Pregunta 10

Escala de Likert	Frecuencia (f)	Porcentaje (%)
1. Totalmente en desacuerdo	18	36 %
2. En desacuerdo	9	18 %
3. Neutral	7	14 %
4. De acuerdo	6	12 %
5. Totalmente de acuerdo	10	20 %
Total	**50**	**100 %**

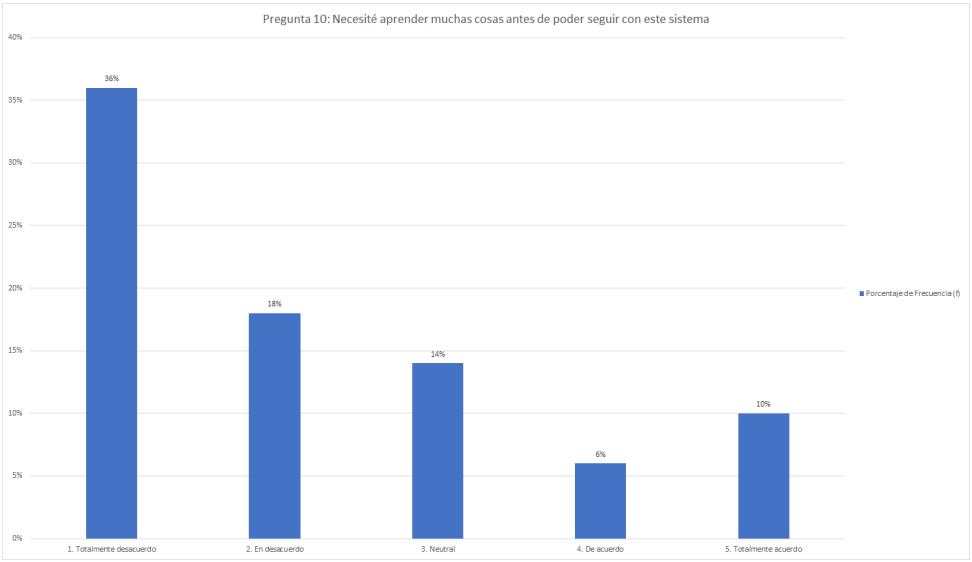


Figura 29.
Distribución de respuestas para la Pregunta 10.

Análisis: Por fin, el 54 % de la muestra (27 personas) niegan necesitar conocimientos técnicos, pre-

vios complejos para operar la plataforma. Por lo tanto, esto refuerza la accesibilidad del diseño frontend, posibilitando a cualquier estudiante sin importar su conocimiento técnico usar de inmediato sus funciones de evaluación.

4.2.4 Calificación Global de Usabilidad y Aceptabilidad

Una vez analizadas las respuestas de forma individual, se procedió al cálculo del puntaje global de la Escala de Usabilidad del Sistema (SUS). Siguiendo la metodología de Brooke, se normalizaron los valores para obtener una puntuación en un rango de 0 a 100.

En la **Tabla CIII** se presentan los estadísticos descriptivos finales derivados de la evaluación de los 50 usuarios.

Tabla CIII: Resultados globales de la evaluación SUS

Indicador Estadístico	**Valor Obtenido**
Muestra efectiva (n)	50
Puntaje Promedio SUS	74.70
Mediana de la Muestra	80.00
Desviación Estándar	16.54
Valor Mínimo Detectado	37.50
Valor Máximo Detectado	100.00
Grado de Usabilidad	**B (Muy Bueno)**
Rango de Aceptabilidad	**Aceptable**

Para contextualizar el desempeño del sistema *PreServerless*, se utiliza el marco de referencia de Bangor et al. (2009) y Sauro (2011), el cual establece una jerarquía lógica entre el puntaje, el grado y su interpretación cualitativa, como se detalla en la **Tabla CIV**.

Tabla CIV: Interpretación lógica de puntajes SUS (Adaptado de Bangor et al., 2009)

Rango SUS	Grado	Adjetivo	Aceptabilidad
$> 80{,}3$	A	Excelente (Superior)	Aceptable
71.4 - 80.3	**B**	**Muy Bueno**	**Aceptable**
51.7 - 71.3	C	Bueno (Promedio)	Marginal
40.9 - 51.6	D	Regular (Pobre)	Marginal
$< 40{,}9$	F	No Aceptable	Inaceptable

Interpretación Final: El puntaje promedio de **74.70** sitúa a la plataforma por encima del umbral de corte estándar de 68 puntos. De acuerdo con la lógica de la **Tabla CIV**, el sistema se clasifica con un Grado **B**, lo que corresponde a una valoración de **Muy Bueno** y un rango de **Aceptable**. Es importante resaltar que la mediana de **80.00** indica que la mitad de los usuarios percibió una usabilidad que roza el grado de excelencia (A), consolidando a *PreServerless* como una solución altamente competitiva y eficaz para el usuario final.

4.2.5 Validación de Fiabilidad del Instrumento (Alfa de Cronbach)

Como paso previo al análisis de los impactos del proyecto, es imperativo validar la consistencia interna del instrumento aplicado. Para ello, se empleó el coeficiente **Alfa de Cronbach** (α), el cual mide la correlación entre los reactivos del SUS para asegurar que la medición de usabilidad sea estable y libre de errores aleatorios.

La ecuación matemática que rige este estadístico se define a continuación:

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k}\sigma_i^2}{\sigma_t^2}\right) \tag{4.3}$$

Donde los componentes del cálculo se definen como:

- α: Coeficiente de fiabilidad (rango de 0 a 1).
- k: Número total de reactivos o preguntas evaluadas ($k = 10$).
- σ_i^2: Varianza observada en cada ítem individual.
- $\sum \sigma_i^2$: Sumatoria de las varianzas de los 10 ítems analizados.
- σ_t^2: Varianza de la puntuación total acumulada por los 50 participantes.

Los resultados obtenidos tras el procesamiento de la muestra de 50 usuarios se detallan en la **Tabla CV**.

Tabla CV: Resultados del análisis de fiabilidad de la muestra

Variable de Control	**Valor Estadístico**
Cantidad de Reactivos (k)	10
Suma de Varianzas de Ítems ($\sum \sigma_i^2$)	12.1747
Varianza de los Puntajes Totales (σ_t^2)	46.5976
Resultado Coeficiente Alfa (α)	**0.821**

Justificación Técnica: El coeficiente 0,821 denota una buena consistencia interna - excelente. Según Hernández-Lalinde et al. Superando este umbral (2018) y la obtención de una medida superior a 0.80, el estudio tiene la suficiente validez científica. Esta robustez estadística deja sin necesidad de las repetitivas aplicaciones (test-retest). Por consiguiente, el impacto descrito debajo es basado sobre alta precisión de datos.

4.3 Análisis de Impactos del Proyecto

Para completar la investigación y el desarrollo de este proyecto, se realizó un análisis de impacto en base al criterio del tesista. Esta evaluación cubre las áreas generales que influyen positiva o negativamente

en el entorno, basándose en la metodología propuesta por Posso. Para estructurar este análisis, se creó una tabla referencial con la escala de valores que se detalla a continuación.

Tabla CVI: Valores de referencia para el análisis de impacto

Valor	Descripción del Impacto
-3	Impacto alto negativo
-2	Impacto medio negativo
-1	Impacto bajo negativo
0	No hay impacto
1	Impacto bajo positivo
2	Impacto medio positivo
3	Impacto alto positivo

Elaboración: Propia. **Fuente:** Posso [53].

4.3.1 Impacto Económico

La evaluación de este impacto puede cuantificar la optimización de los recursos financieros a favor de la Institución cuando se aplica el modelo de Pay As You Go (pago por uso).

Tabla CVII: Matriz de Impacto Económico

Indicador	-3	-2	-1	0	1	2	3	Total
Reducción de costos de mantenimiento operativo							X	3

Eliminación de inversión inicial en hardware	X	3
Eficiencia financiera por el modelo de pago por uso	X	3
Suma Total		**9**

Elaboración: Propia. **Fuente:** Posso [53].

Nivel de Impacto Económico = 9 / 3 = 3 (Alto positivo).

Análisis: Usar servicios gestionados hace a la Universidad no consumir presupuesto en periodos de inactividad académica (ej. vacacional). Debido al hecho de que no necesitas tener servidores físicos funcionando constantemente, la barrera económica de mantenimiento de la plataforma educativa se minimiza mucho.

Justificación de la Ineficiencia de la Arquitectura Tradicional (Monolítica)

Para dimensionar el impacto económico real, es necesario contrastar la propuesta frente al estándar tradicional de alojamiento web mediante servidores virtuales privados o instancias monolíticas en la nube. Un servidor estándar con capacidad mínima para soportar a 50 usuarios concurrentes (por ejemplo, una instancia Amazon EC2 t3.medium) tiene una tarifa oficial aproximada de $0.0416 USD por hora. Al operar de forma ininterrumpida las 730 horas promedio que tiene un mes, el costo base se sitúa en aproximadamente $30.00 USD mensuales, independientemente de si los estudiantes utilizan el sistema o no [54].

Los informes de optimización de infraestructura demuestran que las plataformas con horarios de uso definidos presentan un índice de capacidad ociosa o desperdicio que oscila entre el 60 por ciento y el 80 por ciento del tiempo facturado [55]. En la Tabla CVIII se detalla la comparativa técnica y financiera.

Tabla CVIII: Comparativa de Eficiencia: Monolito vs. PreServerless

Parámetro de Costo	**Sistema Monolítico (Amazon EC2)**	**Arquitectura PreServerless**
Costo Operativo Mensual [54]	$30.00 USD mensuales promedio (Calculado a 730 horas de uso ininterrumpido).	$0.00 USD mensuales (Pago variable fraccionado estrictamente al milisegundo).
Capacidad Ociosa (Desperdicio) [55]	Alto. El servidor genera facturación durante las noches, fines de semana y vacaciones.	Nulo. Si no existen estudiantes conectados, el consumo de cómputo escala a cero.
Mantenimiento de Sistema Operativo	Requiere horas de ingeniería para aplicar parches de seguridad y actualizaciones.	Gestionado íntegramente por el proveedor de nube, eliminando costos de mantenimiento.
Conclusión Económica	**Genera un costo de oportunidad negativo por subutilización constante.**	**Maximiza el retorno de inversión eliminando el gasto por inactividad.**

Elaboración: Propia. **Fuentes:** Cast AI [55], Amazon Web Services [54].

Justificación Técnica Matemática del Gasto Cero:

Al final, la puntuación excelente en impacto económico conseguida en la matriz original de inicio está basada en un tratamiento matemático y contrastan entre consumo plan-piloto y las políticas oficiales de la capa gratuita (Free Tier) de AWS [56]. En el período de validación, los 50 usuarios produjeron menos de 5,000 peticiones a la BD así también a las funciones lambda. Esto sucede debido a que la capa gratuita puede absorber hasta 1,000,000 invocaciones mensuales y sin coste alguno, por lo tanto, sólo consumió un 0.5 % del límite permitido.

La Tabla CIX muestra datos oficiales donde la arquitectura propuesta consume totalmente sin generar facturación la carga operativa del piloto.

Tabla CIX: Comparativa de Capa Gratuita AWS vs. Consumo del Proyecto

Microservicio	**Límite Gratuito Mensual AWS**	**Consumo del Piloto**	**Costo**
AWS Lambda [57]	1,000,000 de peticiones y 400,000 GB-s de cómputo.	Menos de 5,000 peticiones	$0.00
Amazon DynamoDB [56]	25 GB de almacenamiento y 25 WCU/RCU permanentes.	Menos de 0.1 GB de datos	$0.00
API Gateway [56]	1,000,000 de llamadas (HTTP/REST).	Menos de 5,000 llamadas	$0.00
Amazon Cognito [58]	50,000 Usuarios Activos Mensuales (MAU).	50 Usuarios Activos	$0.00
Total Facturado			**$0.00**

Elaboración: Propia. **Fuentes:** Amazon Web Services [56], [57], [58].

4.3.2 Impacto Socio Cultural

Se evalúa la transformación en el proceso de preparación del aspirante universitario y la democratización del acceso a herramientas de calidad.

Tabla CX: Matriz de Impacto Socio Cultural

Indicador	-3	-2	-1	0	1	2	3	Total
Accesibilidad universal desde cualquier dispositivo						X		2
Reducción de la ansiedad del estudiante							X	3
Fomento del estudio autónomo y la competencia sana							X	3
Suma Total								**8**

Elaboración: Propia. **Fuente:** Posso [53].

Nivel de Impacto Socio Cultural = 8 / 3 = 2.67 (Alto positivo).

Análisis: Esta plataforma cierra la desigualdad tecnológica proporcionando un medio de preparación rápido y eficaz para móviles con pocos recursos disponibles. Pero aún más, este sistema de medallas y posiciones anima a los estudiantes a trabajar mejor por sí mismos, disminuyendo la frustración y la ansiedad ante los exámenes de ingreso.

4.3.3 Impacto Tecnológico y Ambiental

A diferencia de los centros de datos locales que consumen electricidad de forma ininterrumpida, se valora la eficiencia energética y operativa de la arquitectura implementada en la nube.

Tabla CXI: Matriz de Impacto Tecnológico y Ambiental

Indicador	-3	-2	-1	0	1	2	3	Total
Reducción del consumo energético (Cómputo bajo demanda)							X	3
Optimización en el uso de recursos de hardware						X		2
Suma Total								**5**

Elaboración: Propia. **Fuente:** Posso [53].

Nivel de Impacto Tecnológico y Ambiental = 5 / 2 = 2.50 (Alto positivo).

Análisis: Como parte del procesamiento se delega a los proveedores de nubes públicas, estamos utilizando una infraestructura compartida de hiperescala que utiliza mucho menos potencia eléctrica y enfriamiento que los servidores propios. Esto da como resultado una menor huella de carbono relacionada con el funcionamiento del sistema escolar.

4.3.4 Impacto Ético y Legal

Se evalúa el cumplimiento de normas con relación al manejo de la información del usuario y a la transparencia algorítmica de calificación.

Tabla CXII: Matriz de Impacto Ético y Legal

Indicador	-3	-2	-1	0	1	2	3	Total
Privacidad y seguridad en la gestión de credenciales							X	3
Transparencia e imparcialidad en la evaluación							X	3
Suma Total								**6**

Elaboración: Propia. **Fuente:** Posso [53].

Nivel de Impacto Ético y Legal = 6 / 2 = 3.00 (Alto positivo).

Análisis: Integración de protocolos de autorización estandarizados: la información confidencial como las contraseñas y los datos de los alumnos nunca quedan expuestos. Por lo demás el motor automatizado de evaluación elimina todo tipo de sesgo del evaluador humano en la calificación de las pruebas, garantizando completa imparcialidad hacia todos los aspirantes.

4.3.5 Nivel de Impacto Global del Proyecto

Para concluir el análisis, se consolida el promedio de todas las áreas evaluadas. Este valor representa la efectividad integral del proyecto dentro del entorno universitario.

Nivel de Impacto Global = Promedio de las cuatro áreas = (3.00 + 2.67 + 2.50 + 3.00) / 4 = 2.79

A continuación, se resume la calificación final del proyecto basándose en los resultados matemáticos obtenidos en las matrices anteriores.

Tabla CXIII: Resumen del Nivel de Impacto Global

Área de Impacto	Promedio	Calificación
Económico	3.00	Alto Positivo
Socio Cultural	2.67	Alto Positivo
Tecnológico y Ambiental	2.50	Alto Positivo
Ético y Legal	3.00	Alto Positivo
TOTAL GLOBAL	**2.79**	**Alto Positivo**

Elaboración: Propia. **Fuente:** Posso [53].

Análisis Final: El Proyecto tiene un Impacto Global de 2.79, dejándolo matemáticamente dentro del rango de Impacto Alto Positivo. Esta métrica fundamentada en el criterio de un experto, se establece que la aplicación de esta arquitectura resuelve el problema técnico inicial y genera un beneficio directo, de forma social, económica e infraestructural institucional.

CONCLUSIONES

- Se desarrolló e implementó satisfactoriamente la aplicación web educativa gamificada Serverless, para proporcionar a los aspirantes universitarios una herramienta tecnológica moderna y accesible. La integración de servicios en la nube permitió transformar el modelo tradicional de preparación preuniversitaria, ofreciendo una plataforma escalable y adaptable a las necesidades actuales de evaluación académica.

- Se consolidó un marco conceptual que permitió validar la viabilidad de la computación serverless y de los enfoques de gamificación educativa a partir de una revisión bibliográfica sistemática y rigurosa. Este sustento teórico sirvió como base para el diseño de una arquitectura funcional compuesta por microservicios orientados a eventos, facilitando la transición desde esquemas monolíticos tradicionales hacia una infraestructura Cloud más flexible, escalable y con alta disponibilidad. Asimismo, la adopción del modelo de pago por uso permitió optimizar recursos operativos y ajustar dinámicamente el consumo de servicios según la demanda de la plataforma.

- Se evaluó la usabilidad de PreServerless mediante la aplicación del instrumento SUS, el cual obtuvo un coeficiente Alfa de Cronbach de 0.821, valor que certificó una fiabilidad excelente. Los resultados evidenciaron un puntaje promedio de 74.70 y una mediana de 80.00, situando a la aplicación en un Grado B (Muy Bueno) y confirmando que la interfaz resulta intuitiva y adecuada para los usuarios finales.

- La arquitectura funcional se diseñó para integrar, nativamente, servicios de AWS tales como Lambda, DynamoDB, Cognito y el API gateway. Respecto al Nivel de impacto global se logró Alcanzar un Nivel de Impacto Global de 2.79 (Alto Positivo) al mantener el coste en el pilotaje a 0.00USD. Esto resultó en una notable reducción económica frente a los que bordean alrededor de 30.00 USD/mes, dato necesario para mantener un servidor tradicional

RECOMENDACIONES

- Institucionalizar el uso de PreServerless como herramienta central de evaluación para los aspirantes universitarios, garantizando que la plataforma se consolide como el eje de preparación académica desde el inicio de los ciclos de estudio, con el fin de maximizar el aprovechamiento de sus recursos y funcionalidades.

- Mantener una actualización permanente del marco conceptual y tecnológico de la plataforma. Debido a la rápida evolución de los servicios de computación en la nube, se recomienda investigar e incorporar progresivamente nuevas tendencias y metodologías de enseñanza digital basadas en tecnologías cloud.

- Configurar Alarmas de facturación y Límites de Consumo en la Consola de AWS usando un enfoque FinOps. Si bien la arquitectura creada posibilitó conservar un déficit nulo en la etapa piloto, es necesario supervisar la utilización de recursos frente a un aumento multitudinario de estudiantes para asegurar la viabilidad económica del sistema.

- Al lograr evolucionar el desarrollo de software gracias a la integración de algoritmos de IA en las funciones del backend, la plataforma debe adaptarse para no sólo seleccionar los reactivos en forma aleatoria, sino también integrar a cada alumno la pregunta de la dificultad según su rendimiento histórico.

- Auditar con una base centrada en la experiencia y la UI del cuestionario SUS cuando se exhibieron tendencias netas neutrales, provoca el mayor refinamiento de estos elementos visuales y permite elevar la mediana de uso arriba del 80.3 a una escala de rango Grade A (Excelente), tomando en cuenta las futuras versiones que sucedan en la plataforma.

BIBLIOGRAFÍA

[1] M. Li, S. Ma e Y. Shi, "Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis," *Frontiers in Psychology*, vol. 14, pág. 1 253 549, 2023. DOI: 10.3389/fpsyg.2023.1253549

[2] E. Y. M. Olvera, E. G. J. Bastidas y G. J. M. Espinoza, "Análisis de la brecha digital y el acceso a recursos tecnológicos en las instituciones de educación secundaria en Ecuador," *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, n.º 2, págs. 6697-6719, 2024. DOI: 10.37811/cl_rcm.v8i2.11086

[3] Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), *Proyecto de inversión: Sistema Nacional de Nivelación y Admisión*, dic. de 2020. dirección: https://www.educacionsuperior.gob.ec/wp-content/uploads/2022/02/PERFIL_PROYECTO_SNNA_.pdf

[4] H. B. Hassan, S. A. Barakat y Q. I. Sarhan, "Survey on serverless computing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 10, n.º 1, pág. 39, 2021. DOI: 10.1186/s13677-021-00253-7

[5] E. Jonas, J. Schleier-Smith, V. Sreekanti et al., *Cloud programming simplified: A Berkeley view on serverless computing*, arXiv preprint arXiv:1902.03383, 2019. DOI: 10.48550/arXiv.1902.03383

[6] Amazon Web Services, *AWS Lambda – Developer Guide*, Amazon Web Services, Inc., feb. de 2025. dirección: https://docs.aws.amazon.com/pdfs/lambda/latest/dg/lambda-dg.pdf

[7] Amazon Web Services, *Amazon DynamoDB – Developer Guide*, Amazon Web Services, Inc., feb. de 2025. dirección: https://docs.aws.amazon.com/pdfs/amazondynamodb/latest/developerguide/dynamodb-dg.pdf

[8] G. Lampropoulos y A. Sidiropoulos, "Impact of gamification on students' learning outcomes and academic performance: A longitudinal study comparing online, traditional, and gamified learning," *Education Sciences*, vol. 14, n.º 4, pág. 367, 2024. DOI: 10.3390/educsci14040367

[9] A. Ortiz, "Architecting serverless microservices on the cloud with AWS," en *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Association for Computing Machinery, 2019, págs. 1240-1240. DOI: 10.1145/3287324.3287533

[10] J. Brooke, "SUS: A quick and dirty usability scale," en *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. Weerdmeester e I. L. McClelland, eds., London: Taylor & Francis, 1996, págs. 189-194.

[11] Valparaíso News, "PreuApp: Innovación en la preparación preuniversitaria en Chile," *Valparaíso News*, pág. 37, feb. de 2025. dirección: https://www.litoralpress.cl/SimbiuPDF/2025/02/14/5712311.pdf

[12] S. A. Bello, L. O. Oyedele, O. O. Akinade et al., "Cloud computing in construction industry: Use cases, benefits and challenges," *Automation in Construction*, vol. 122, pág. 103 441, 2021. DOI: 10.1016/j.autcon.2020.103441

[13] P. Pierleoni, R. Concetti, A. Belli y L. Palma, "Amazon, Google and Microsoft solutions for IoT: Architectures and a performance comparison," *IEEE Access*, vol. 8, págs. 5455-5470, 2019. DOI: 10.1109/ACCESS.2019.2961511

[14] M. Zbořil y V. Svatá, "Performance comparison of cloud virtual machines," *Journal of Systems and Information Technology*, vol. 27, n.º 2, págs. 197-213, 2025. DOI: 10.1108/JSIT-02-2022-0040

[15] A. N. Toosi, B. Javadi, A. Iosup, E. Smirni y S. Dustdar, "Serverless computing for next-generation application development," *Future Generation Computer Systems*, vol. 164, págs. 1-5, 2025. DOI: 10.1016/j.future.2024.107573

[16] E. N. Okwuduba, K. C. Nwosu, E. C. Okigbo, N. N. Samuel y C. Achugbu, "Impact of intrapersonal and interpersonal emotional intelligence and self-directed learning on academic performance among pre-university science students," *Heliyon*, vol. 7, n.º 3, e06611, 2021. DOI: 10.1016/j.heliyon.2021.e06611

[17] O. Karpenko y V. Ostapchuk, "Institutions of professional higher education as a condition for successful personality development," *Social Work and Education*, vol. 10, n.º 2, págs. 224-231, 2023. DOI: 10.25128/2520-6230.23.2.6

[18] T. Luca, A. Zamfiroiu, A. S. Ghai y E. M. Ciupercă, "E-learning services integration for pre-university education system," en *Proceedings of the 19th International Conference on Virtual Learning (ICVL)*, vol. 19, 2024, págs. 23-34. DOI: 10.58503/icvl-v19y202402

[19] S. G. A. van Herpen, M. Meeuwisse, W. H. A. Hofman y S. E. Severiens, "A head start in higher education: The effect of a transition intervention on interaction, sense of belonging, and academic performance," *Studies in Higher Education*, vol. 45, n.º 4, págs. 862-877, 2020. DOI: 10.1080/03075079.2019.1572088

[20] C. E. S. Bautista, C. Figueroa y P. A. S. Cubides, "Acercamiento teórico al concepto de tecnología desde la educación en tecnología," *Boletín Redipe*, vol. 10, n.º 5, págs. 110-120, 2021.

[21] R. J. A. Guerrero, Á. E. M. Simisterra y J. D. O. Ponce, "Impacto de nuevas tecnologías en la educación universitaria en Ecuador," *Qualitas Revista Científica*, vol. 23, n.º 23, págs. 12-21, 2021. DOI: 10.55867/qual23.02

[22] P. Jaiswal, "Integrating educational technologies to augment learners' academic achievements," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, n.º 2, págs. 145-159, 2020. DOI: 10.3991/ijet.v15i02.11809

[23] T. An y M. Oliver, "What in the world is educational technology? Rethinking the field from the perspective of the philosophy of technology," *Learning, Media and Technology*, vol. 46, n.º 1, págs. 6-19, 2020. DOI: 10.1080/17439884.2020.1810066

[24] J. Chen y M. Liang, "Play hard, study hard? The influence of gamification on students' study engagement," *Frontiers in Psychology*, vol. 13, pág. 994 700, 2022. DOI: 10.3389/fpsyg.2022.994700

[25] M. Kalogiannakis, S. Papadakis y A. I. Zourmpakis, "Gamification in science education: A systematic review of the literature," *Education Sciences*, vol. 11, n.º 1, pág. 22, 2021. DOI: 10.3390/educsci11010022

[26] J. Krath, L. Schürmann y H. F. O. von Korflesch, "Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious

games and game-based learning," *Computers in Human Behavior*, vol. 125, pág. 106 963, 2021. DOI: 10.1016/j.chb.2021.106963

[27] C. Baah, I. Govender y P. R. Subramaniam, "Exploring the role of gamification in motivating students to learn," *Cogent Education*, vol. 10, n.º 1, pág. 2 210 045, 2023. DOI: 10.1080/2331186X.2023.2210045

[28] K. Palaniappan y N. M. Noor, "Gamification strategy to support self-directed learning in an online learning environment," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, n.º 3, págs. 104-116, 2022. DOI: 10.3991/ijet.v17i03.27489

[29] T. C. Ion y E. Popescu, "An innovative distance learning platform for mathematics education in secondary schools: Design, development and preliminary studies," *Education and Information Technologies*, vol. 30, n.º 5, págs. 5529-5560, 2025. DOI: 10.1007/s10639-024-13040-z

[30] R. A. Salas-Rueda y C. Alvarado-Zamorano, "Design of creative virtual spaces through the use of a web application during the educational process about bank savings," *Creativity Studies*, vol. 15, n.º 2, págs. 299-315, 2022. DOI: 10.3846/cs.2022.12304

[31] B. Agbo, C. Morris, M. Osman, J. Basketts y T. Kyriacou, "A systematic literature review on software applications used to support curriculum development and delivery in primary and secondary education," *International Journal of Educational Research Open*, vol. 7, pág. 100 385, 2024. DOI: 10.1016/j.ijedro.2024.100385

[32] E. A. Fırat y S. Fırat, "Web 3.0 in learning environments: A systematic review," *Turkish Online Journal of Distance Education*, vol. 22, n.º 1, págs. 148-169, 2020. DOI: 10.17718/tojde.849898

[33] J. Alamari y C. E. Chow, "A proposed design for cross platform applications using web technologies," *International Journal of Computer Applications*, vol. 174, n.º 29, págs. 41-46, 2021. DOI: 10.5120/ijca2021921220

[34] J. Vepsäläinen, M. Hevery y P. Vuorimaa, "Resumability: A new primitive for developing web applications," *IEEE Access*, vol. 12, págs. 9038-9046, 2024. DOI: 10.1109/ACCESS.2024.3352891

[35] S. Criollo-C et al., "Towards the integration of emerging technologies as support for the teaching and learning model in higher education," *Sustainability*, vol. 15, n.º 7, pág. 6055, 2023. DOI: 10.3390/su15076055

[36] P. Bocciarelli, A. D'Ambrogio, T. Panetti y A. Giglio, "E-MDAV: A framework for developing data-intensive web applications," *Informatics*, vol. 9, n.º 1, pág. 12, 2022. DOI: 10.3390/informatics9010012

[37] J. M. Ríos y N. P. Souto, "Comparison of development methodologies in web applications," *Information and Software Technology*, vol. 119, pág. 106 238, 2019. DOI: 10.1016/j.infsof.2019.106238

[38] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou y Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, vol. 8, n.º 3, págs. 685-698, 2022. DOI: 10.1109/TBDATA.2020.2975587

[39] N. Chaplynska, S. Ivanov, S. Onishchenko, A. Chukhlib e I. Tkach, "Trends in the development of web computing in e-commerce," *Periodicals of Engineering and Natural Sciences*, vol. 13, n.º 1, págs. 233-248, 2025. DOI: 10.21533/pen.v13.i1.250

[40] N. Sakimura, J. Bradley y N. Agarwal, "Proof key for code exchange by OAuth public clients," Internet Engineering Task Force (IETF), RFC 7636, sep. de 2015. DOI: 10.17487/RFC7636

[41] A. Parecki, D. Waite y T. Brodin, "OAuth 2.0 for browser-based apps," Internet Engineering Task Force (IETF), Internet-Draft draft-ietf-oauth-browser-based-apps-15, oct. de 2023.

[42] D. Hardt, "The OAuth 2.0 Authorization Framework," Internet Engineering Task Force (IETF), RFC 6749, oct. de 2012. DOI: 10.17487/RFC6749

[43] P. C. et al., “Advancements in authentication mechanisms using OAuth 2.0 and SAML,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, n.º 6, 2024. DOI: 10.56726/IRJMETS66276

[44] C. Verwijs y D. Russo, “A theory of Scrum team effectiveness,” *ACM Transactions on Software Engineering and Methodology*, vol. 32, n.º 3, 74:1-74:51, 2023. DOI: 10.1145/3571849

[45] A. Quintas-Hijós, C. Peñarrubia-Lozano y J. C. Bustamante, “Analysis of the applicability and utility of a gamified didactics with exergames at primary schools: Qualitative findings from a natural experiment,” *PLOS ONE*, vol. 15, n.º 4, e0231269, 2020. DOI: 10.1371/journal.pone.0231269

[46] H. Edison, X. Wang y K. Conboy, “Comparing methods for large-scale agile software development: A systematic literature review,” *IEEE Transactions on Software Engineering*, vol. 48, n.º 8, págs. 2709-2731, 2022. DOI: 10.1109/TSE.2021.3069039

[47] A. Alami y O. Krancher, “How Scrum adds value to achieving software quality?” *Empirical Software Engineering*, vol. 27, n.º 7, pág. 165, 2022. DOI: 10.1007/s10664-022-10208-4

[48] J. N. da Silva Junior, M. S. C. Teotônio, A. J. M. L. Junior, J. G. B. V. Pinheiro y L. L. da Silva, “Gamification 2.0: Gamifying an entire introductory organic chemistry course again,” *Journal of Chemical Education*, vol. 102, n.º 2, págs. 679-687, 2025. DOI: 10.1021/acs.jchemed.4c01183

[49] C. Redondo-Rodríguez, J. A. Becerra-Mejías, G. Gil-Fernández y F. J. Rodríguez-Velasco, “Influence of gamification and cooperative work in peer, mixed and interdisciplinary teams on emotional intelligence, learning strategies and life goals that motivate university students to study,” *International Journal of Environmental Research and Public Health*, vol. 20, n.º 1, pág. 547, 2023. DOI: 10.3390/ijerph20010547

[50] C. J. de Parga, *UML: Aplicaciones en Java y C++*. Madrid, España: Ra-Ma, 2011, Disponible en la Biblioteca Digital UTN, ISBN: 9788499640822.

[51] D. Castilla, I. Jaen, C. Suso-ribera et al., “Psychometric properties of the Spanish full and short forms of the System Usability Scale (SUS): Detecting the effect of negatively worded items,” *International Journal of Human-Computer Interaction*, vol. 40, n.º 15, págs. 4145-4151, 2023. DOI: 10.1080/10447318.2023.2209840

[52] A. Bangor, P. T. Kortum y J. T. Miller, “An empirical evaluation of the System Usability Scale,” *International Journal of Human-Computer Interaction*, vol. 24, n.º 6, págs. 574-594, 2008. DOI: 10.1080/10447310802205776

[53] M. Á. P. Yépez, *Proyectos, tesis y marco lógico. Planes e informes de investigación*. Ibarra, Ecuador: Miguel Ángel Posso Yépez, 2011, ISBN: 978-9942-03-930-9.

[54] Amazon Web Services. “Amazon EC2 pricing: On-demand instances,” visitado 12 de mayo de 2026. dirección: https://aws.amazon.com/ec2/pricing/on-demand/

[55] Cast AI Framework, “Cloud cost optimization: Kubernetes benchmark and idle capacity report,” Cast AI Analytics, 2024. dirección: https://cast.ai/blog/cloud-cost-optimization/

[56] Amazon Web Services. “AWS free tier: Cloud computing services,” visitado 12 de mayo de 2026. dirección: https://aws.amazon.com/free/

[57] Amazon Web Services. “AWS Lambda pricing: Pay only for what you use,” visitado 12 de mayo de 2026. dirección: https://aws.amazon.com/lambda/pricing/

[58] Amazon Web Services. “Amazon Cognito pricing,” visitado 12 de mayo de 2026. dirección: https://aws.amazon.com/cognito/pricing/

ANEXOS

4.4 Anexo A.- Formato y Métricas del Instrumento de Evaluación (Google Forms)

La documentación del proceso de recolección de datos requiere una exposición detallada del instrumento tal como fue percibido por los sujetos de prueba, así como la evidencia de su aplicación. Para evitar ambigüedades metodológicas y permitir una revisión exhaustiva, se desglosa el cuestionario *System Usability Scale* (SUS) visualizando la identidad del formulario, la estructura de los reactivos psicométricos y, finalmente, las métricas de participación generadas por la plataforma. Esta disposición garantiza la trazabilidad metodológica en la evaluación de la plataforma *PreServerless*.

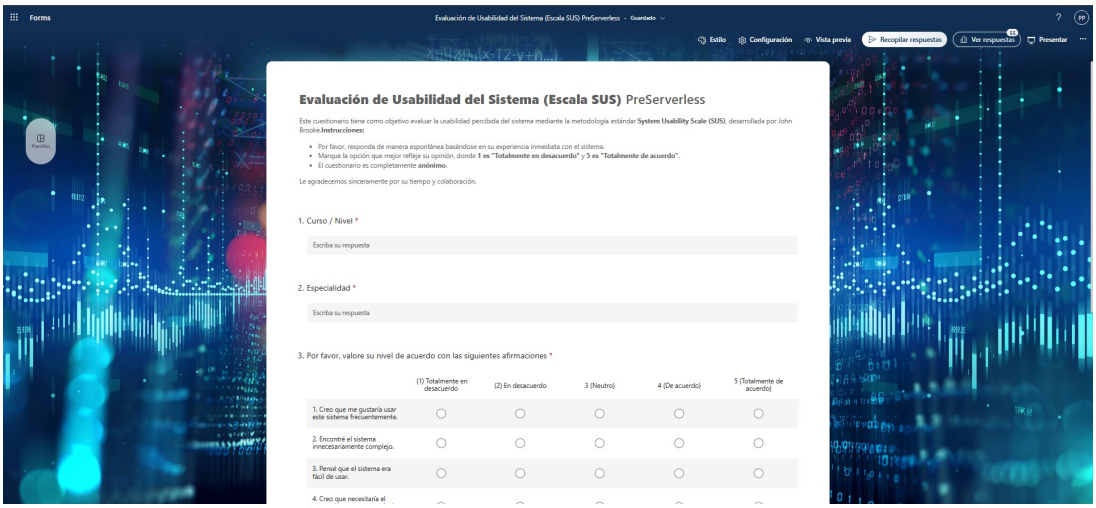


Figura 30.
Encabezado del formulario, objetivos de la investigación e instrucciones para el usuario.

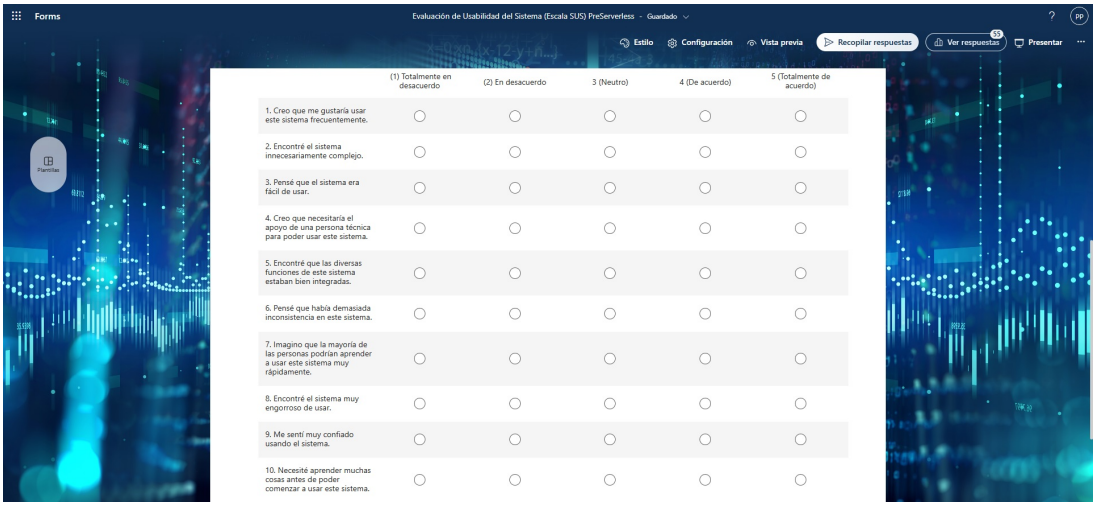


Figura 31.
Sección de reactivos de usabilidad empleando la escala Likert de cinco puntos.

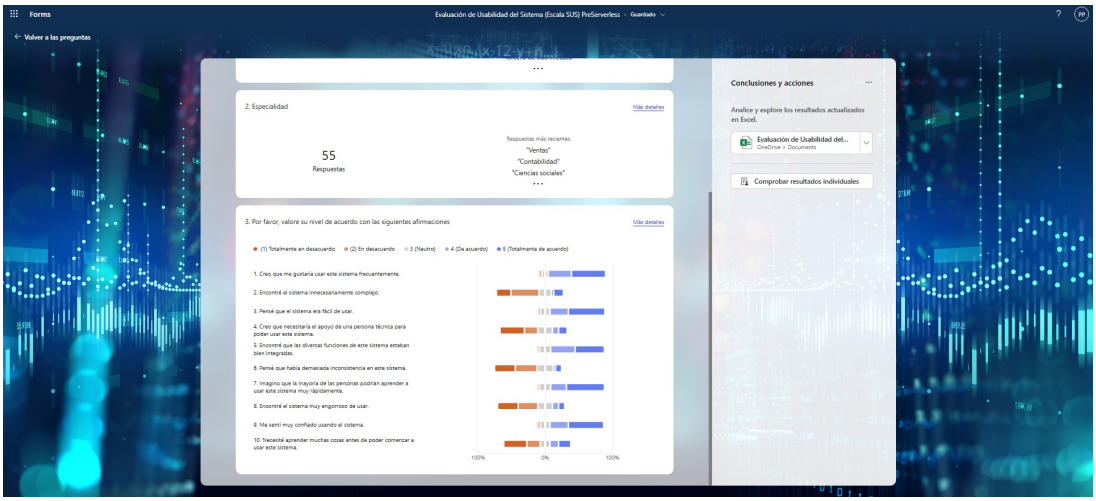


Figura 32.

Métricas de recolección de datos y resumen automatizado de participación generadas por la herramienta.

4.5 Anexo B.- Diagrama de Interacción Orientado a Eventos (FaaS)

En este anexo se detalla la lógica de interacción a nivel de código para la plataforma *PreServerless*. A diferencia de los diagramas topológicos generales, esta representación de secuencia modela el flujo de datos bajo una arquitectura orientada a eventos (*Event-Driven Architecture*). Se describe cómo las peticiones del usuario actúan como desencadenadores (*triggers*) que invocan funciones como servicio (FaaS) en AWS Lambda, ejecutadas en el entorno Node.js.

Asimismo, se ilustra la implementación de métodos específicos del *AWS SDK for JavaScript*, tales como *QueryCommand* para la recuperación dinámica de reactivos y *UpdateItemCommand* para la persistencia de puntajes y métricas de gamificación en Amazon DynamoDB. Esta documentación técnica permite auditar el comportamiento asíncrono y la eficiencia en la gestión de estados de la aplicación sin recurrir a servidores persistentes.

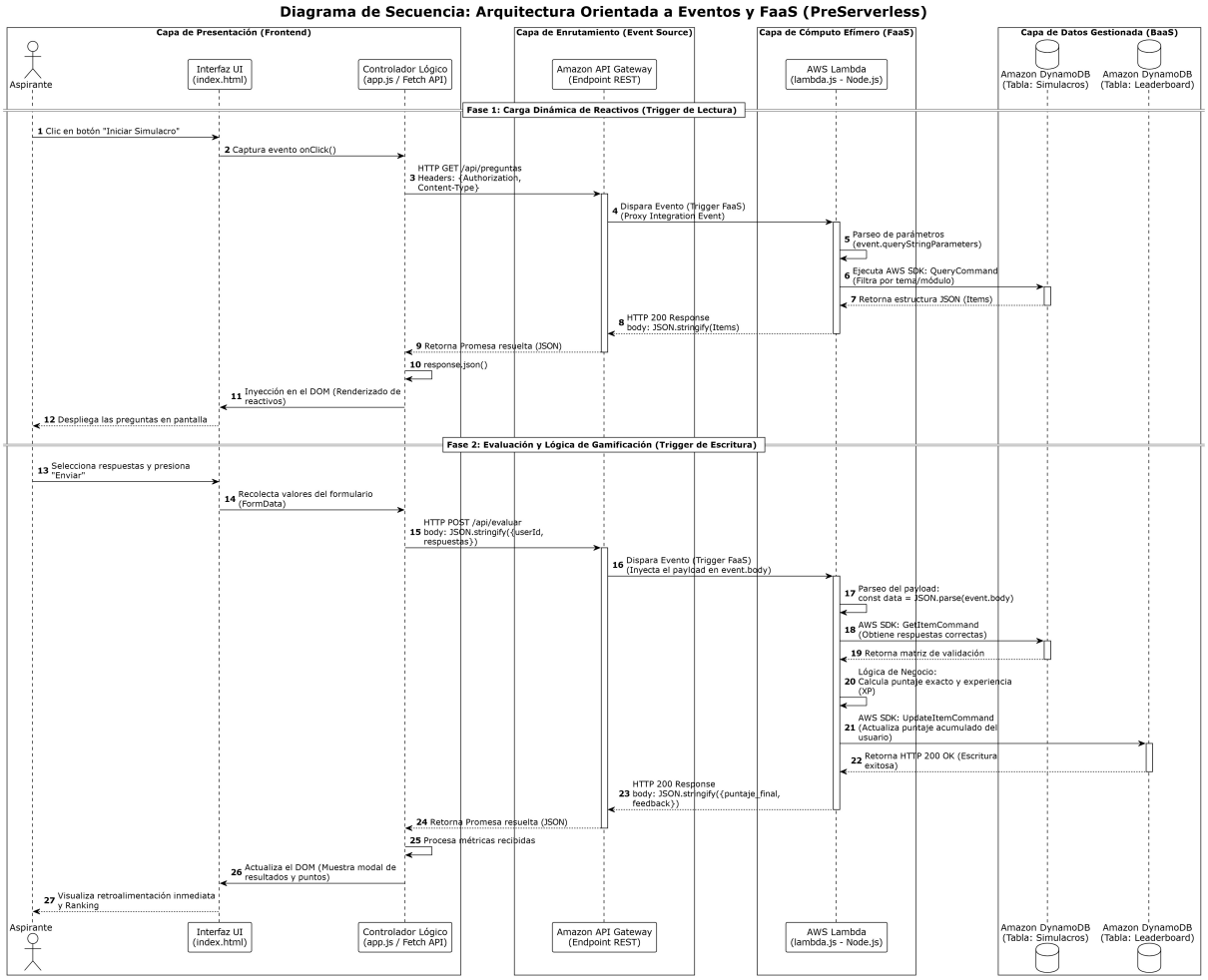


Figura 33.

Diagrama de secuencia de invocaciones FaaS y lógica de negocio en la infraestructura AWS.

4.6 Anexo C.- Interfaces de la Aplicación *PreServerless*

En este anexo se presenta la evidencia visual del ecosistema *PreServerless* funcionando en producción. Se documenta el flujo completo desde el acceso público, la gestión administrativa de datos y la experiencia interactiva del estudiante.

4.6.1 C.1. Módulo de Gestión Administrativa

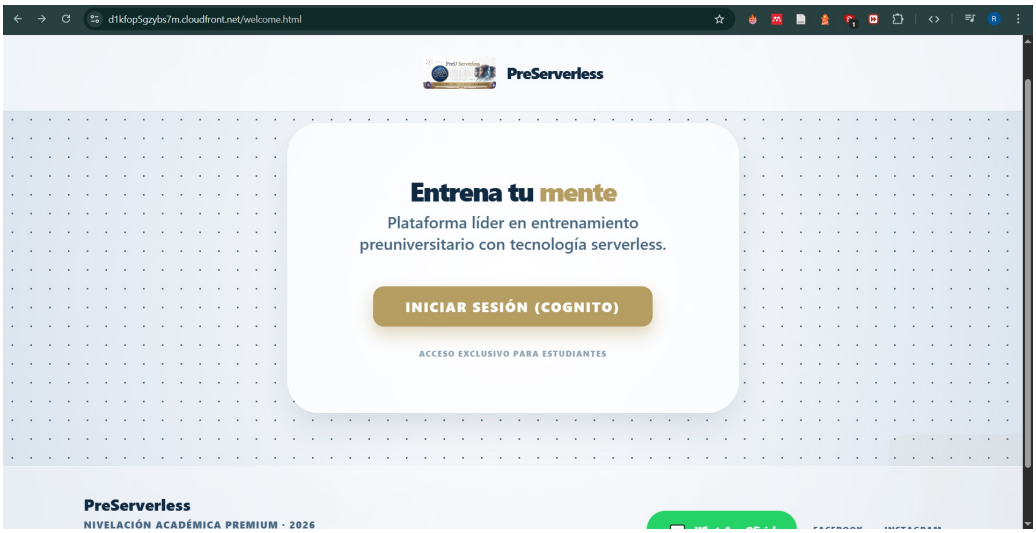


Figura 34.
Acceso inicial al sistema.



Figura 35.
Portal de autenticación y control de seguridad gestionado por AWS Cognito.

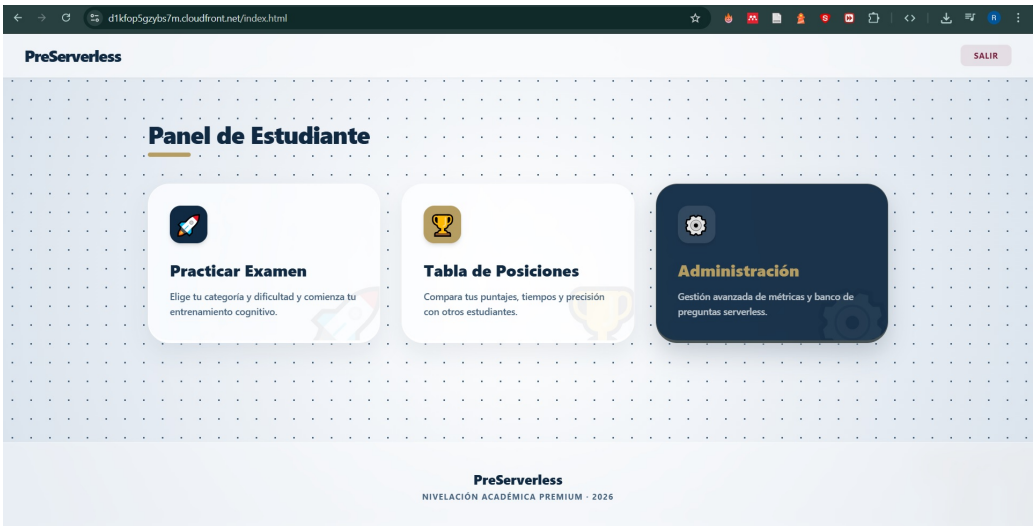


Figura 36.

Panel de navegación principal con control de acceso basado en roles (RBAC).

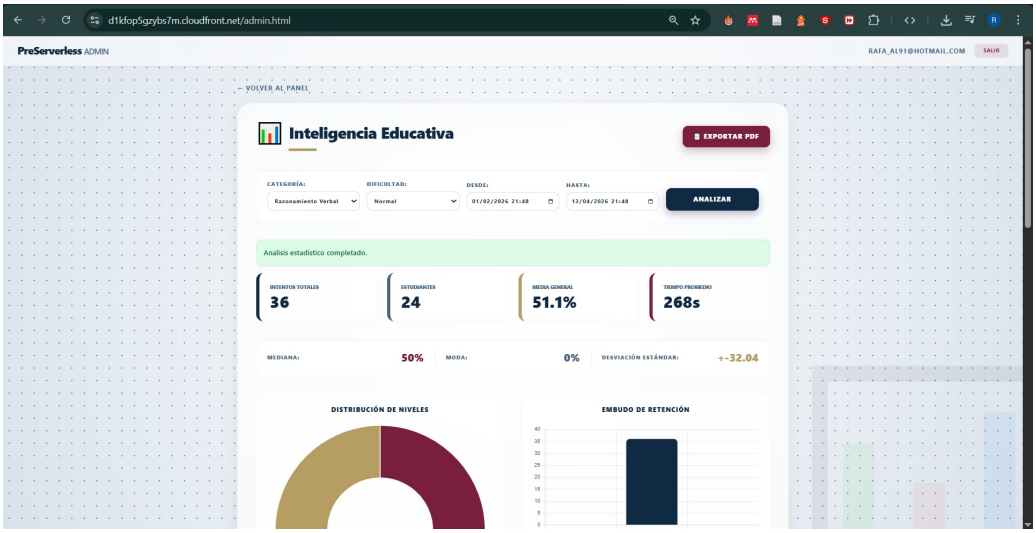


Figura 37.

Módulo de Inteligencia Educativa para el análisis estadístico de rendimientos.

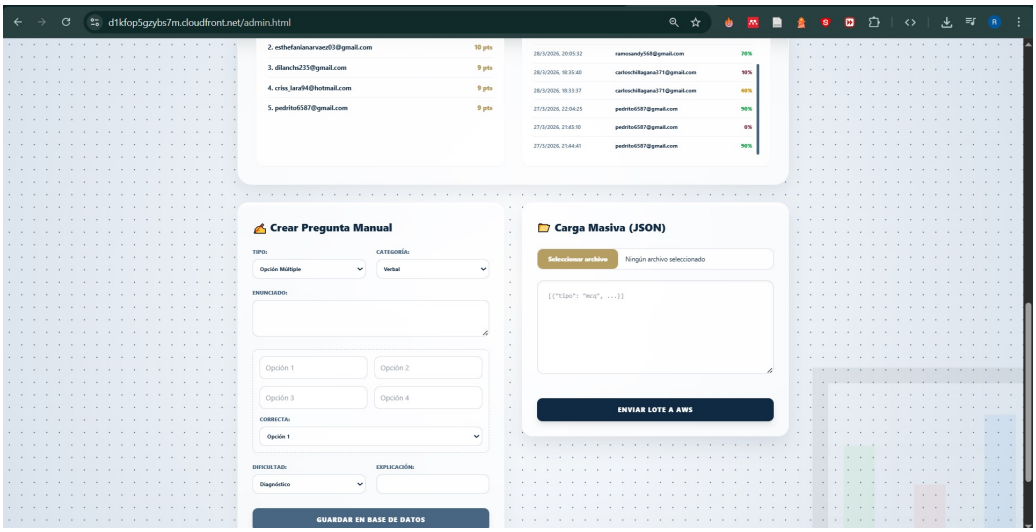


Figura 38.
Gestión de persistencia NoSQL: Carga manual y masiva de reactivos mediante JSON.

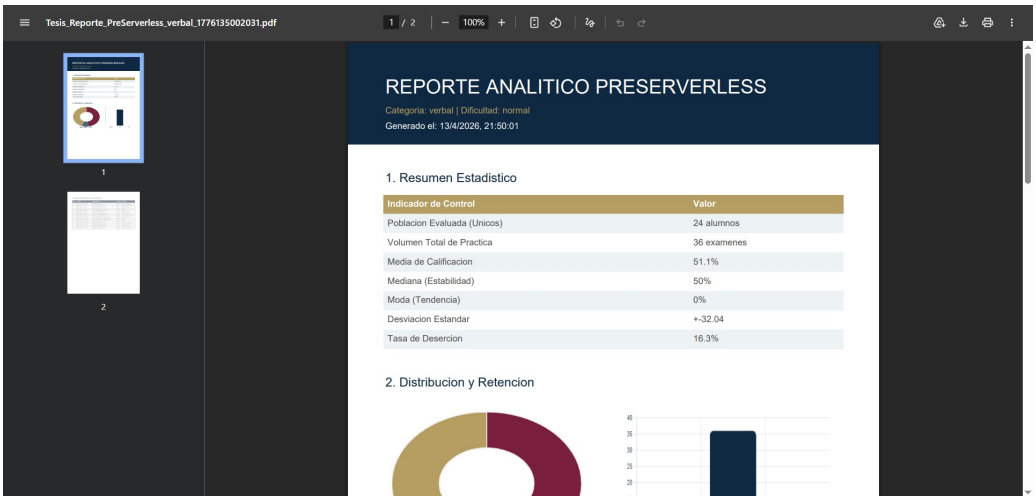


Figura 39.
Documento de reporte analítico consolidado generado automáticamente en formato PDF.

4.6.2 C.2. Módulo del Estudiante (Evaluación y Gamificación)

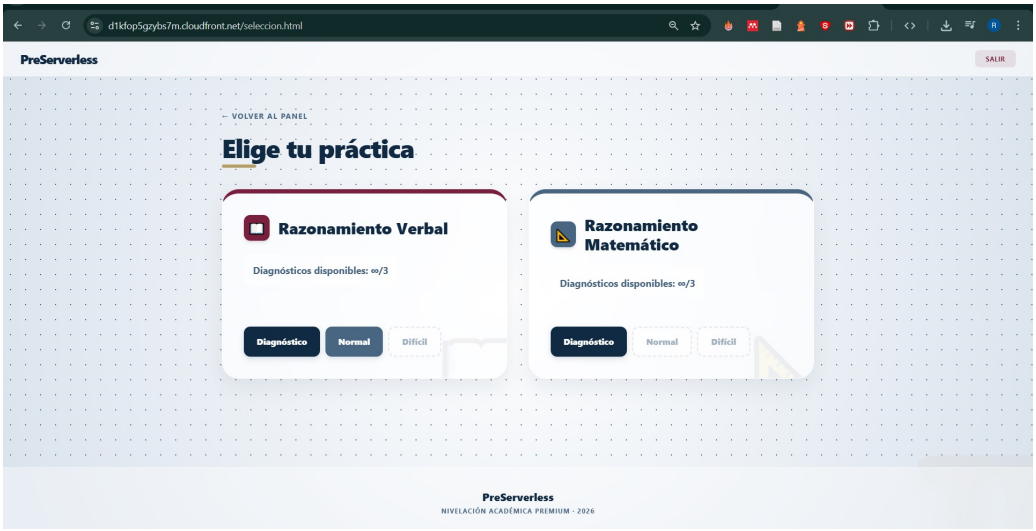


Figura 40.

Selección de áreas de conocimiento y diagnóstico inicial para el aspirante.

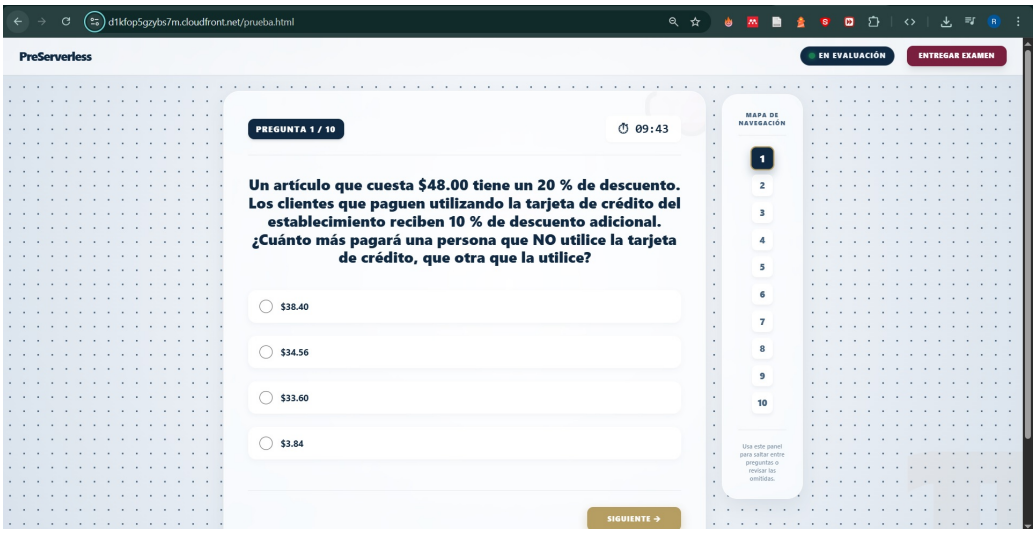


Figura 41.

Entorno de ejecución del simulacro con carga dinámica de reactivos y control de tiempo.

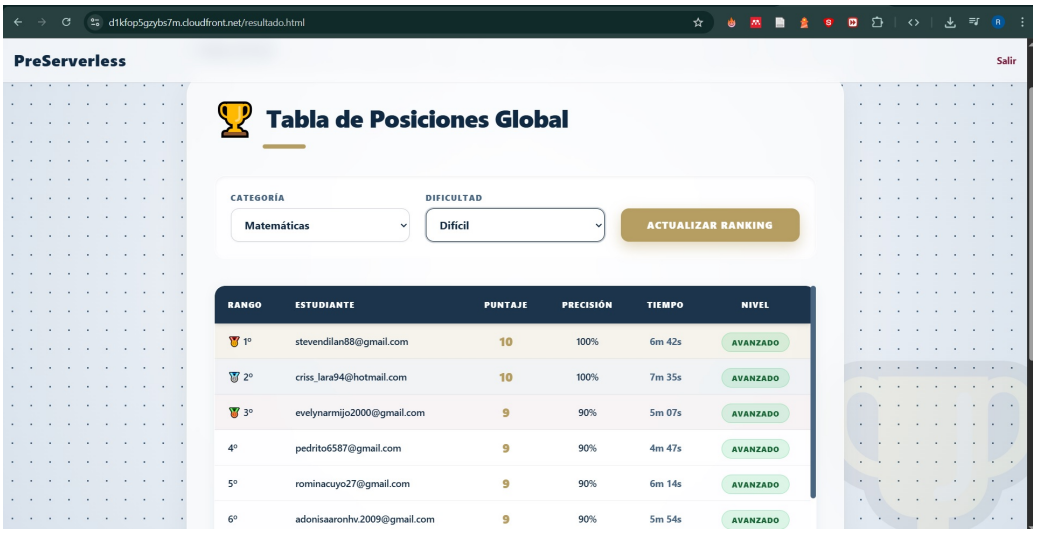


Figura 42.

Visualización del Ranking Global y mecánicas de gamificación (*Leaderboard*).

4.7 Anexo D.- Matriz de Tabulación y Validación Estadística

En este anexo se detalla el procesamiento estadístico para determinar la consistencia interna del instrumento SUS. La validación se realizó mediante el coeficiente Alfa de Cronbach, procesando los datos de la muestra ($N = 50$) de forma automatizada para garantizar la precisión de los resultados.

4.7.1 D.1. Evidencia de Procesamiento en Entorno de Desarrollo

Se utilizó el lenguaje de programación **Python** para la normalización de los ítems de polaridad invertida y el cálculo de varianzas. A continuación, se presenta la captura del script ejecutado en *Visual Studio Code*, donde se evidencia la salida del coeficiente final.

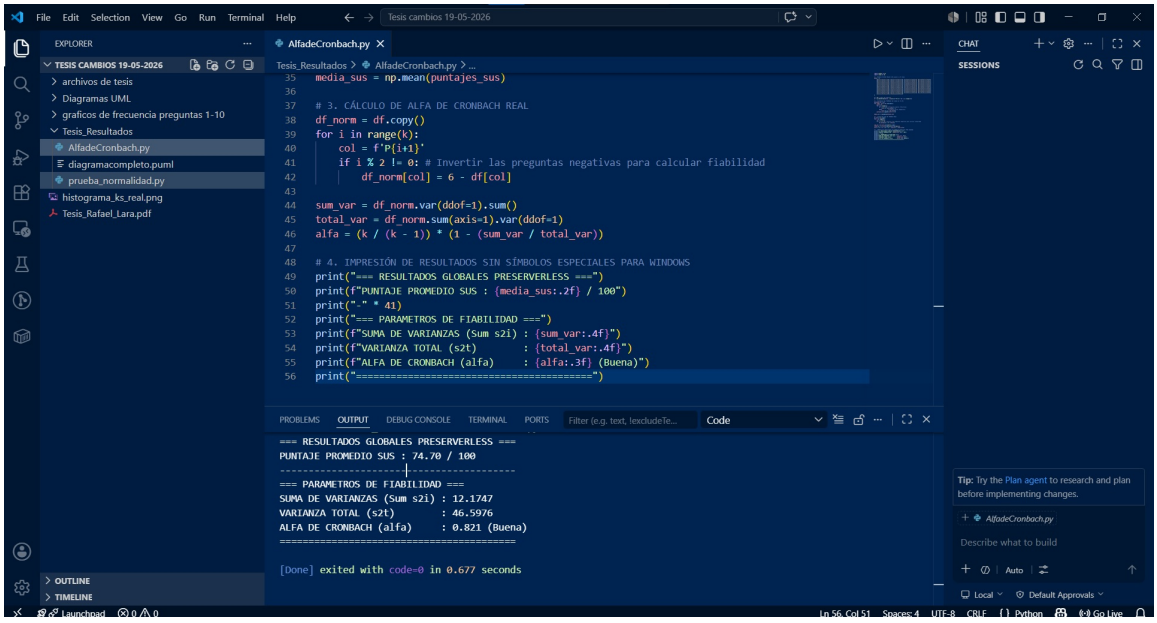


Figura 43.
Script de automatización estadística y salida en consola (Alfa = 0.821).

4.7.2 D.2. Resumen Matemático de Fiabilidad

Para el cálculo del coeficiente se aplicó la siguiente relación matemática:

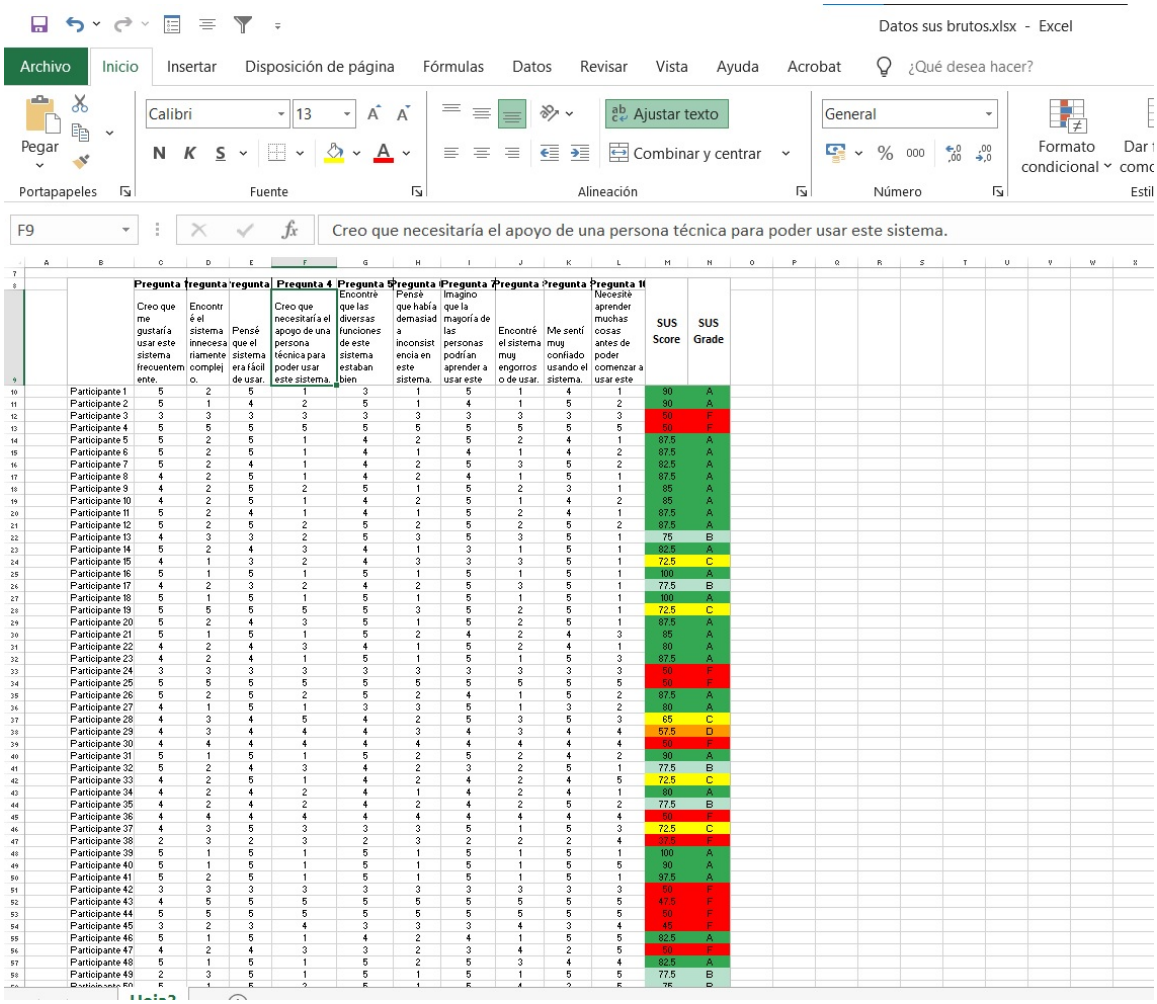
$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum s_i^2}{s_t^2}\right)$$

Los parámetros obtenidos mediante el procesamiento del script se resumen en la siguiente tabla:

Parámetro Estadístico	**Valor Obtenido**
Número de ítems evaluados (k)	10
Sumatoria de varianzas por ítem ($\sum s_i^2$)	12.1747
Varianza de las puntuaciones totales (s_t^2)	49.5976
Coeficiente Alfa de Cronbach (α)	**0.821**

4.7.3 D.3. Matriz de Datos Consolidada (Exportación)

Finalmente, se presenta la matriz de datos crudos exportada a formato de hoja de cálculo, la cual sirvió de base para el análisis presentado en el Capítulo IV.



	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10	SUS Score	SUS Grade
	Creo que me gustaría usar este sistema frecuentemente.	Encontré el sistema innecesariamente complejo.	Pensé que el sistema era fácil de usar.	Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema.	Encontré que las diversas funciones de este sistema estaban bien	Pensé que había demasiada inconsistencia en este sistema.	Imagino que la mayoría de las personas podrían aprender a usar este	Encontré el sistema muy engorroso de usar.	Me sentí muy confiado usando el sistema.	Necesité aprender muchas cosas antes de poder comenzar a usar este		
Participante 1	5	2	5	1	3	1	5	1	4	1	90	A
Participante 2	5	1	4	2	5	1	4	1	5	2	90	A
Participante 3	3	3	3	3	3	3	3	3	3	3	50	F
Participante 4	5	5	5	5	5	5	5	5	5	5	50	F
Participante 5	5	2	5	1	4	2	5	2	4	1	87.5	A
Participante 6	5	2	5	1	4	1	4	1	4	2	87.5	A
Participante 7	5	2	4	1	4	2	5	3	5	2	82.5	A
Participante 8	4	2	5	1	4	2	4	1	5	1	87.5	A
Participante 9	4	2	5	2	5	1	5	2	3	1	85	A
Participante 10	4	2	5	1	4	2	5	1	4	2	85	A
Participante 11	5	2	4	1	4	1	5	2	4	1	87.5	A
Participante 12	5	2	5	2	5	2	5	2	5	2	87.5	A
Participante 13	4	3	3	2	5	3	5	3	5	1	75	B
Participante 14	5	2	4	3	4	1	3	1	5	1	82.5	A
Participante 15	4	1	3	2	4	3	3	3	5	1	72.5	C
Participante 16	5	1	5	1	5	1	5	1	5	1	100	A
Participante 17	4	2	3	2	4	2	5	3	5	1	77.5	B
Participante 18	5	1	5	1	5	1	5	1	5	1	100	A
Participante 19	5	5	5	5	5	3	5	2	5	1	72.5	C
Participante 20	5	2	4	3	5	1	5	2	5	1	87.5	A
Participante 21	5	1	5	1	5	2	4	2	4	3	85	A
Participante 22	4	2	4	3	4	1	5	2	4	1	80	A
Participante 23	4	2	4	1	5	1	5	1	5	3	87.5	A
Participante 24	3	3	3	3	3	3	3	3	3	3	50	F
Participante 25	5	5	5	5	5	5	5	5	5	5	50	F
Participante 26	5	2	5	2	5	2	4	1	5	2	87.5	A
Participante 27	4	1	5	1	3	3	5	1	3	2	80	A
Participante 28	4	3	4	5	4	2	5	3	5	3	65	C
Participante 29	4	3	4	4	4	3	4	3	4	4	57.5	D
Participante 30	4	4	4	4	4	4	4	4	4	4	50	F
Participante 31	5	1	5	1	5	2	5	2	4	2	90	A
Participante 32	5	2	4	3	4	2	3	2	5	1	77.5	B
Participante 33	4	2	5	1	4	2	4	2	4	5	72.5	C
Participante 34	4	2	4	2	4	1	4	2	4	1	80	A
Participante 35	4	2	4	2	4	2	4	2	5	2	77.5	B
Participante 36	4	4	4	4	4	4	4	4	4	4	50	F
Participante 37	4	3	5	3	3	3	5	1	5	3	72.5	C
Participante 38	2	3	2	3	2	3	2	2	2	4	37.5	F
Participante 39	5	1	5	1	5	1	5	1	5	1	100	A
Participante 40	5	1	5	1	5	1	5	1	5	5	90	A
Participante 41	5	2	5	1	5	1	5	1	5	1	97.5	A
Participante 42	3	3	3	3	3	3	3	3	3	3	50	F
Participante 43	4	5	5	5	5	5	5	5	5	5	47.5	F
Participante 44	5	5	5	5	5	5	5	5	5	5	50	F
Participante 45	3	2	3	4	3	3	3	4	3	4	45	F
Participante 46	5	1	5	1	4	2	4	1	5	5	82.5	A
Participante 47	4	2	4	3	3	2	3	4	2	5	50	F
Participante 48	5	1	5	1	5	2	5	3	4	4	82.5	A
Participante 49	2	3	5	1	5	1	5	1	5	5	77.5	B

Figura 44.
Tabulación de respuestas en Excel.