|  |
| --- |
|  |
| ATAM: Explorando el Universo |
| *Proyecto: Explorando el Universo* |
|  |
| **Revisión*: 1.0*** |
| **20/01/2025** |

|  |
| --- |
|  |

Índice

[**Ficha del documento 3**](#_heading=h.1fob9te)

[**1.ATAM 5**](#_heading=h.tyjcwt)

[Acciones 5](#_heading=h.3dy6vkm)

[Resultados Esperados 6](#_heading=h.1ksv4uv)

[**2.Objetivos de negocio 7**](#_heading=h.1t3h5sf)

[**3.Arquitectura 8**](#_heading=h.4d34og8)

[**4.Enfoque arquitectónico 10**](#_heading=h.2s8eyo1)

[**5.Árbol de utilidad 12**](#_heading=h.17dp8vu)

[**6.Análisis de enfoque arquitectónico 13**](#_heading=h.26in1rg)

[**7.Brainstorming y Priorización de Escenarios 17**](#_heading=h.lnxbz9)

[**8.Análisis de los Enfoques Arquitectónicos 18**](#_heading=h.35nkun2)

# **Ficha del documento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *20/01/2025* | *01* | *Raquel R* | * *Formato al documento* |
| *21/01/2025* | *02* | *Raquel R* | * *Trabajo en el punto 1, 4 y 5* |

Documento validado por las partes en fecha:

|  |  |  |
| --- | --- | --- |
| Por el cliente |  | Por la empresa suministradora |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

# 

# 1.ATAM

Este documento tiene como objetivo analizar las decisiones arquitectónicas tomadas para el proyecto **"Explorando el Universo"** y su influencia en los aspectos clave de calidad del sistema. "Explorando el Universo" es una plataforma de aprendizaje en línea diseñada para ofrecer cursos de astronomía a estudiantes de niveles básico y medio. Su propósito es fomentar la curiosidad y el interés en la astronomía mediante contenido interactivo, evaluaciones, foros de discusión y recursos multimedia.

Para evaluar las decisiones arquitectónicas del sistema, utilizaremos el **método ATAM** (Architecture Tradeoff Analysis Method), una técnica que permite identificar riesgos y compromisos, además de analizar cómo estas decisiones impactan el desempeño y las características no funcionales de la plataforma. ATAM se implementa a través de las siguientes cuatro fases:

1. **Identificación de Atributos de Calidad**: En esta fase, se identifican los atributos de calidad que son más críticos para el éxito del sistema. En el caso de "Explorando el Universo", los atributos priorizados incluyen:
   * **Seguridad**: Para proteger la información personal de los usuarios.
   * **Escalabilidad**: Para asegurar que la plataforma pueda soportar un aumento en el número de usuarios.
   * **Disponibilidad**: Para garantizar que la plataforma esté accesible las 24 horas del día.
2. **Escenarios de Uso**: Se crean escenarios representativos que describen cómo los usuarios interactuarán con el sistema. Por ejemplo, un escenario relevante sería la **inscripción de un estudiante en un curso**, que debe ser fluido y sin errores. Este escenario podría incluir:
   * La navegación a través de la lista de cursos.
   * La selección de un curso y proceso de inscripción.
   * La confirmación y acceso al contenido del curso.
3. **Identificación de Riesgos y Compromisos**: En esta etapa, se analizan los riesgos asociados con las decisiones arquitectónicas y los compromisos que deben hacerse. Algunos riesgos identificados incluyen:
   * **Cuellos de botella en la capa de datos**: A medida que crece el número de usuarios, el tiempo de respuesta del sistema puede verse afectado si no se toma en cuenta la optimización de base de datos.
   * **Interdependencias entre componentes**: La arquitectura modular puede ser compleja; si un componente falla, puede afectar la totalidad del sistema.
4. **Evaluación de Decisiones**: Aquí se discuten las decisiones arquitectónicas en relación con los atributos de calidad y los riesgos identificados. Por ejemplo:
   * La elección de **PostgreSQL** para la gestión de datos se justifica debido a su capacidad para manejar grandes volúmenes de datos y asegurar la integridad de la información. A su vez, se implementarán mecanismos de caché para evitar cuellos de botella.
   * Las medidas de **seguridad**, como el uso de **JWT y OAuth 2.0**, no solo cumplen con los requisitos de protección de datos, sino que también ofrecen una experiencia de usuario sin fricciones durante la autenticación.

# **2.Objetivos de negocio**

Los objetivos del negocio para la plataforma "Explorando el Universo" están diseñados para asegurar una experiencia de aprendizaje efectiva, accesible y segura. A continuación se describen los objetivos clave, junto con sus métricas de éxito:

**1. Accesibilidad Universal**

* **Objetivo**: Garantizar que la plataforma sea completamente accesible para todos los usuarios, incluyendo personas con discapacidades visuales y auditivas.
  + **Métrica de Éxito**: Cumplimiento con los estándares **WCAG 2.1** Nivel AA, con informes de auditoría trimestrales para verificar la accesibilidad.
* **Objetivo**: Asegurar la compatibilidad con una amplia gama de dispositivos (computadoras, tabletas y teléfonos inteligentes) y navegadores web populares.
  + **Métrica de Éxito**: Realización de pruebas de usabilidad en diferentes navegadores y dispositivos, con una tasa de satisfacción del usuario superior al 85%.

**2. Experiencia de Usuario Personalizada**

* **Objetivo**: Proporcionar a los estudiantes un acceso rápido y sencillo a sus cursos, permitiéndoles retomar su progreso justo donde lo dejaron mediante enlaces directos al contenido más reciente.
  + **Métrica de Éxito**: Reducción del tiempo medio de inicio de sesión y acceso al curso en un 20% en comparación con versiones anteriores de la plataforma.
* **Objetivo**: Facilitar una navegación intuitiva y la comprensión del contenido a través de una interfaz clara y organizada.
  + **Métrica de Éxito**: Alcanzar una puntuación de usabilidad (SUS - System Usability Scale) superior a 80 en evaluaciones realizadas con usuarios.

**3. Disponibilidad Continua**

* **Objetivo**: Asegurar que la plataforma educativa esté disponible las 24 horas del día, permitiendo el acceso a los contenidos en cualquier momento y lugar.
  + **Métrica de Éxito**: Mantener una disponibilidad del sistema del 99.9%, con monitoreo constante y registros de tiempo de inactividad.
* **Objetivo**: Establecer horarios específicos para la comunicación con los profesores, garantizando el soporte académico dentro de esos períodos.
  + **Métrica de Éxito**: Recibir comentarios positivos en un 90% sobre la accesibilidad y efectividad de la comunicación con el personal docente.

**4. Seguridad de la Información**

* **Objetivo**: Implementar medidas de seguridad robustas, como cifrado de datos en tránsito y en reposo, controles de acceso y autenticación multifactorial (MFA).
  + **Métrica de Éxito**: Realización de auditorías de seguridad semestrales con resultados que confirmen la eficacia de las medidas de seguridad implementadas.

**5. Escalabilidad y Mantenimiento Eficiente**

* **Objetivo**: Diseñar la arquitectura del sistema de manera modular para facilitar futuras actualizaciones, nuevas funcionalidades y correcciones de errores sin interrumpir la operatividad existente.
  + **Métrica de Éxito**: Capacidad para implementar nuevas características en un plazo máximo de 4 semanas, conservando un tiempo de inactividad mínimo.
* **Objetivo**: Asegurar un código bien documentado y estructurado para optimizar el tiempo de desarrollo y mantenimiento.
  + **Métrica de Éxito**: Mantener una cobertura de pruebas de al menos el 85% en todas las nuevas funciones implementadas.

**6. Fomento del Aprendizaje Interactivo**

* **Objetivo**: Ofrecer recursos multimedia enriquecidos, como videos educativos con subtítulos, simulaciones interactivas y foros de discusión, para mejorar la experiencia de aprendizaje.
  + **Métrica de Éxito**: Aumentar la participación de los estudiantes en foros de discusión en un 30% en comparación con ciclos anteriores.
* **Objetivo**: Proporcionar herramientas de evaluación efectivas que permitan a los estudiantes medir su progreso en los distintos módulos del curso.
  + **Métrica de Éxito**: Lograr que el 75% de los estudiantes complete las evaluaciones al finalizar los módulos.

**7. Cumplimiento de Regulaciones y Estándares**

* **Objetivo**: Asegurar que la plataforma cumpla con las normativas de accesibilidad web y seguridad de datos, evitando riesgos legales y garantizando la inclusión de todos los usuarios.
  + **Métrica de Éxito**: Completar auditorías trimestrales de cumplimiento y obtener resultados satisfactorios en todas las áreas evaluadas.

# 

## **3. Arquitectura**

# La arquitectura del sistema "Explorando el Universo" se basa en un modelo de cuatro capas que garantiza escalabilidad, seguridad y mantenibilidad. Este modelo está compuesto por las siguientes capas:

# **3.1. Capa de Presentación (Frontend)**

# **Componentes:**

# **Aplicación Web Responsiva**: Desarrollada con HTML, CSS y JavaScript. Esta capa es responsable de la presentación del contenido educativo, proporcionando una experiencia de usuario intuitiva y fluida.

# **Interfaz Accesible**: La interfaz de usuario seguirá los estándares **WCAG 2.1** Nivel AA, asegurando que las personas con discapacidades visuales y auditivas puedan interactuar con el contenido. Esto implicará el uso de subtítulos en videos y un diseño que garantice un adecuado contraste entre texto y fondo.

# **Integración con Servicios de Terceros**: Se integrarán servicios externos para gestionar foros de discusión, sistemas de comentarios y notificaciones, mejorando la interacción del usuario.

# **Función:**

# **Gestión del Inicio de Sesión**: Permitir el inicio de sesión de estudiantes y profesores, asegurando el acceso mediante autenticación multifactorial.

# **Gestión de Perfiles**: Ofrecer opciones para que los usuarios modifiquen y eliminen su información de perfil.

# **Búsqueda e Inscripción en Cursos**: Facilitar la búsqueda de cursos y la inscripción directa, así como proporcionar acceso al contenido y a las evaluaciones.

# **Presentación de Certificados**: Generar y mostrar certificados de finalización una vez que los cursos han sido aprobados.

# **Contribución a los Objetivos del Negocio:**

# Mejora la experiencia del usuario a través de una interfaz intuitiva y accesible.

# Garantiza la compatibilidad con dispositivos móviles y escritorio, permitiendo un acceso flexibles a la plataforma.

# Permite el manejo de un alto número de usuarios simultáneos sin degradar el rendimiento, asegurando disponibilidad continua.

# **3.2. Capa de Negocios (Modelos)**

# **Componentes:**

# **Modelos de Django**: Se utilizarán los modelos de Django para definir la estructura de datos del sistema, donde cada modelo representará una entidad del sistema, como usuarios, cursos, evaluaciones y comentarios.

# **Validaciones y Reglas de Negocio**: Implementación de validaciones en los modelos para asegurar que los datos sean coherentes y cumplan con las reglas del negocio (por ejemplo, validación de la inscripción en cursos y control de acceso).

# **Función:**

# **Definición de Estructura de Datos**: Establecer relaciones entre entidades y definir campos necesarios para almacenar información relevante.

# **Lógica de Negocio**: Aplicar restricciones y reglas de negocio a los datos, garantizando la integridad y calidad de la información.

# **Contribución a los Objetivos del Negocio:**

# Asegura que la plataforma opere con datos consistentes y válidos, contribuyendo a una mejor experiencia de usuario.

# Permite la escalabilidad del sistema, facilitando la adición de nuevas funcionalidades y la adaptación a requisitos futuros.

# **3.3. Capa de Vistas (Views)**

# **Componentes:**

# **Vistas de Django**: Se implementarán vistas basadas en clases y funciones para gestionar la lógica de presentación, manejando las operaciones que el usuario puede ejecutar, como la inscripción en cursos, la visualización de contenido y la entrega de evaluaciones.

# **Manejo de Peticiones HTTP**: Las vistas recibirán y procesarán las solicitudes HTTP, conectando la capa de presentación con los modelos y asegurando la presentación de datos adecuados.

# **Función:**

# **Gestión de Solicitudes**: Procesar las solicitudes del usuario y devolver las respuestas adecuadas, gestionando la interacción entre el frontend y la lógica de negocio.

# **Control de Flujos**: Asegurar que los usuarios solo accedan a las funcionalidades y datos a los que están autorizados, contribuyendo a la seguridad del sistema.

# **Contribución a los Objetivos del Negocio:**

# Facilita la creación de una experiencia de usuario fluida y continua al gestionar interacciones sin interrupciones.

# Permite la modularidad del sistema, facilitando el mantenimiento y la implementación de nuevas características.

# **3.4. Capa de Acceso a Datos (Base de Datos)**

# **Componentes:**

# **Base de Datos PostgreSQL**: Utilizará **PostgreSQL** como sistema de gestión de bases de datos, permitiendo manejar grandes volúmenes de datos de manera eficiente y ofreciendo funcionalidades avanzadas como consultas complejas y soporte

# **4.Investigación y Análisis**

**4.1 Identificación de enfoques arquitectónicos**

Se han identificado los siguientes enfoques arquitectónicos para el sistema "Explorando el Universo":

1. **Autenticación basada en JWT (JSON Web Token)**
   * Implementación de un sistema de autenticación basado en **JWT**, garantizando sesiones seguras y una gestión eficiente de usuarios.
   * Seguridad mejorada mediante la implementación de **OAuth 2.0** para la autenticación con cuentas institucionales (como Google y Microsoft).
2. **Arquitectura de tres capas**
   * **Capa de Presentación**: Interfaz de usuario desarrollada utilizando HTML, CSS y JavaScript, asegurando una experiencia interactiva.
   * **Capa de Lógica de Negocio**: Manejo de la lógica del sistema utilizando **Django** como framework principal.
   * **Capa de Datos**: Almacenamiento de datos en **PostgreSQL**, permitiendo un acceso eficiente y seguro a la información.
   * Esta estructura facilita la escalabilidad y mantenibilidad del sistema a medida que crece la base de usuarios.
3. **Cifrado de datos sensibles**
   * Implementación del estándar **AES-256** para la protección de información confidencial, como contraseñas y datos personales de estudiantes y profesores.
   * Encriptación en tránsito mediante **SSL/TLS** para garantizar la seguridad en la comunicación de datos.
4. **Notificaciones automáticas**
   * Uso de notificaciones por correo electrónico para recordar fechas importantes como inscripciones, evaluaciones y certificados de finalización.
   * Integración de servicios de notificaciones que aseguran la entrega efectiva y puntual de mensajes a los usuarios.
5. **Modularidad en el código**
   * Separación de funcionalidades clave como:
     + Gestión de cursos y perfiles.
     + Evaluaciones automáticas y manuales.
     + Administración de contenidos y certificados.
   * Esta modularidad facilita la ampliación y el mantenimiento del código.
6. **Integración con infraestructura existente**
   * Conexión con sistemas de autenticación de instituciones para ofrecer una experiencia de usuario fluida y consistente.
   * Sincronización con plataformas educativas existentes para mejorar la gestión del contenido.
7. **Respaldo automático**
   * Implementación de un sistema de respaldo automatizado de la base de datos cada 12 horas para evitar la pérdida de información crítica.
   * Almacenamiento de respaldos en entornos seguros y de fácil acceso para restauraciones rápidas.
8. **Compatibilidad multiplataforma**
   * Desarrollo de la aplicación utilizando tecnologías web responsivas para garantizar accesibilidad desde diferentes dispositivos y navegadores, optimizando la experiencia en computadoras, tabletas y teléfonos inteligentes.
   * Asegurar que la plataforma sea accesible desde diferentes navegadores, incluyendo Chrome, Firefox y Edge.
9. **Tolerancia a fallos**
   * Implementación de estrategias de recuperación ante fallos mediante balanceo de carga para garantizar disponibilidad continua.
   * Monitoreo del sistema con herramientas como **Prometheus** y **Grafana** para detectar incidentes rápidamente y tomar medidas correctivas.
10. **Cumplimiento de estándares de accesibilidad**
    * Diseño de la interfaz conforme a los estándares **WCAG 2.1** Nivel AA, asegurando la inclusión de personas con discapacidades visuales o motoras.
    * Realización de pruebas de usabilidad enfocadas en la accesibilidad para garantizar que el sistema sea usable por todos.
11. **Escalabilidad horizontal**
    * Diseño de una arquitectura que permita el escalado horizontal mediante la adición de nodos según la demanda de usuarios simultáneos.
    * Uso de contenedores con Docker para facilitar el despliegue del sistema en diferentes entornos.

# 5.Árbol de utilidad

En esta sección del documento, se presentarán los escenarios de calidad en forma de un árbol de utilidad.

Imagen que contiene Diagrama

Descripción generada automáticamente

# 

# 6.Análisis de enfoque arquitectónico

**6.1 Enfoques Arquitectónicos Seleccionados**

**Enfoque Arquitectónico Seleccionado: Arquitectura por Capas**

* **Descripción**: La arquitectura por capas divide el sistema en cuatro componentes principales:
  + **Capa de Presentación**: Donde se encuentra la interfaz de usuario y las interacciones directas del usuario.
  + **Capa de Lógica de Negocio**: Que maneja la lógica y las reglas de negocio de la aplicación.
  + **Capa de Acceso a Datos**: Que se encarga de la gestión de la base de datos y la persistencia de los datos.
  + **Capa de Almacenamiento**: Que garantiza la correcta gestión de los datos en la base de datos o sistemas de archivos.

**Evaluación de Factores de Calidad**:

1. **Fiabilidad**:
   * **Impacto**: La arquitectura por capas facilita el manejo de errores en la lógica de negocio, permitiendo una recuperación más sencilla y asegurando que los componentes no se vean afectados entre sí.
2. **Rendimiento**:
   * **Impacto**: La optimización adecuada de la capa de acceso a datos permite una rápida recuperación de información y un manejo eficiente de las solicitudes de los usuarios simultáneos.
3. **Disponibilidad**:
   * **Impacto**: La arquitectura permite implementar soluciones de alta disponibilidad, así como balanceo de carga, asegurando que el sistema esté accesible casi todo el tiempo y cumpliendo con los estándares del negocio.
4. **Mantenibilidad**:
   * **Impacto**: La separación clara de las capas permite que los futuros desarrollos y correcciones de errores se realicen de manera más sencilla, sin riesgo de afectar otras partes del sistema.
5. **Seguridad**:
   * **Impacto**: La implementación de medidas de seguridad en cada capa, como validaciones y controles de acceso, facilita la protección de datos sensibles.

**6.2 Brainstorming y Priorización de Escenarios**

En esta fase, se realizó un **brainstorming** (tormenta de ideas) con los miembros del equipo para identificar y crear una lista de escenarios relevantes relacionados con la arquitectura del sistema. Se discutieron los diferentes escenarios que podrían surgir durante la construcción y el despliegue de la plataforma educativa.

**Escenarios Identificados**:

1. Inscripción de estudiantes en cursos.
2. Registro y autenticación de usuarios.
3. Gestión de contenidos por parte de los profesores.
4. Interacción de estudiantes y profesores en foros de discusión.
5. Recuperación de contraseñas.
6. Acceso a la plataforma desde diferentes dispositivos.
7. Evaluación del rendimiento del sistema bajo carga simulada.

**Priorización de Escenarios**: Se utilizó una clasificación basada en la importancia y el impacto de cada escenario en el sistema. Los escenarios fueron evaluados según la probabilidad de ocurrencia y el impacto en la calidad del sistema:

* **Alta Prioridad**:
  + Inscripción de estudiantes en cursos.
  + Registro y autenticación de usuarios.
* **Media Prioridad**:
  + Gestión de contenidos por parte de los profesores.
  + Evaluación del rendimiento del sistema bajo carga simulada.
* **Baja Prioridad**:
  + Recuperación de contraseñas.
  + Interacción de estudiantes y profesores en foros de discusión.
  + Acceso a la plataforma desde diferentes dispositivos.

# 7.Análisis de los Enfoques Arquitectónicos

**7.1 Enfoque Arquitectónico Seleccionado: Arquitectura por Capas**

**Descripción:** La arquitectura por capas es un patrón de diseño que separa el sistema en distintos niveles, permitiendo una organización clara y el manejo eficiente de las responsabilidades. Este enfoque divide el sistema en las siguientes capas:

1. **Capa de Presentación**:
   * Comprende la interfaz de usuario y todas las interacciones que el usuario puede realizar con la aplicación. Se encarga de la presentación de la información y de captar las acciones del usuario.
2. **Capa de Lógica de Negocio**:
   * Maneja la lógica y las reglas del negocio. Aquí se implementan los procesos y las decisiones del sistema, asegurando que la aplicación opere correctamente y que las reglas de negocio se apliquen adecuadamente.
3. **Capa de Acceso a Datos**:
   * Se encarga de la gestión de la persistencia de datos. Esta capa interactúa con la base de datos, realizando operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y gestionando la consistencia e integridad de los datos.
4. **Capa de Almacenamiento**:
   * Incluye la base de datos o sistemas de archivos donde se almacenan permanentemente los datos. Esta capa garantiza que la información sea duradera y accesible según sea necesario.

**7.2 Evaluación de Factores de Calidad**

1. **Fiabilidad**:
   * **Impacto**: La arquitectura por capas mejora la fiabilidad al facilitar el manejo de errores en la capa de lógica de negocio. Permite que los componentes se recuperen de fallos sin afectar a otras partes del sistema, aumentando así la confianza en el rendimiento del sistema en entornos de producción.
2. **Rendimiento**:
   * **Impacto**: La correcta optimización de la capa de acceso a datos es crucial para el rendimiento. Permite una rápida recuperación de información y asegura un manejo eficiente de las solicitudes de múltiples usuarios, lo que contribuye a que las interacciones con la aplicación sean fluidas y rápidas.
3. **Disponibilidad**:
   * **Impacto**: La arquitectura permite implementar soluciones de alta disponibilidad y balanceo de carga, lo que asegura que el sistema esté accesible prácticamente todo el tiempo, cumpliendo con las expectativas de un entorno educativo activo y demandante.
4. **Mantenibilidad**:
   * **Impacto**: La separación clara de las capas facilita el mantenimiento y la evolución del sistema. Las modificaciones pueden llevarse a cabo en una capa aislada, sin comprometer el funcionamiento general del sistema, lo que es vital para implementar actualizaciones y mejoras continuas.
5. **Seguridad**:
   * **Impacto**: La implementación de medidas de seguridad en cada capa (como validaciones y controles de acceso) contribuye a la protección de datos sensibles. Esta arquitectura permite aplicar y gestionar las políticas de seguridad de manera más efectiva, asegurando que cada capa pueda contener sus propias protecciones.

**7.3 Conclusiones del Análisis**

El análisis de los enfoques arquitectónicos seleccionados para el sistema "Explorando el Universo" resalta las fortalezas de la arquitectura por capas, demostrando que este modelo proporciona una base sólida para garantizar la fiabilidad, rendimiento, disponibilidad, mantenibilidad y seguridad del sistema. Al permitir una gestión clara de las responsabilidades y facilitar futuras ampliaciones o mejoras, este enfoque es fundamental para el éxito y el crecimiento sostenible del proyecto.

Se recomienda seguir manteniendo esta estructura al desarrollar el sistema, asegurando que los principios de diseño se aplican correctamente en la implementación y que se revisan constantemente en función de las necesidades del usuario y los desafíos del entorno.