



Vigilada Mineducación

VTA/LRS: Una aproximación a la implementación de un asistente virtual como herramienta de soporte al proceso de enseñanza

VTA/LRS: An Approach to Implementing a Virtual Assistant as a Support Tool for the Teaching Process

Autor:

LUCAS VERA TORO

Trabajo de grado para optar por el título de Magíster en Ingeniería

Asesor:

JUAN CARLOS MONTOYA MENDOZA

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Maestría en Ingeniería

MEDELLÍN

2023

CONTENIDO

LISTADO DE TABLAS	4
LISTADO DE ILUSTRACIONES	6
RESUMEN	10
ABSTRACT	11
1. INTRODUCCIÓN	12
2. PLANTEAMIENTO DEL PROBLEMA	15
2.1. PREGUNTA DE INVESTIGACIÓN	18
2.2. IDEA DE INVESTIGACIÓN	18
3. JUSTIFICACIÓN	19
4. OBJETIVOS	20
4.1. OBJETIVO GENERAL	20
4.2. OBJETIVOS ESPECÍFICOS	20
5. MARCO TEÓRICO Y ESTADO DEL ARTE	21
5.1. MARCO TEÓRICO Y CONCEPTUAL	21
5.1.1. IDENTIFICAR ATENCIÓN USANDO IMÁGENES	21
5.1.2. LEARNING ANALYTICS	26
5.1.3. ASISTENTES DE ENSEÑANZA	27
5.1.4. ARQUITECTURAS ESCALABLES EN LA NUBE Y SERVERLESS	28
5.1.5. MODELOS DE LENGUAJE GENERATIVOS	35
5.2. ESTADO DEL ARTE	39
6. DESARROLLO DEL TRABAJO	41
6.1. REQUISITOS	42
6.1.1. Requisitos funcionales	42

6.1.2.	Requisitos no funcionales	44
6.2.	ARQUITECTURA DE LA SOLUCIÓN	44
6.2.1.	Casos de uso	47
6.2.2.	Analizador de capturas de pantalla	49
6.2.3.	Asistente del profesor	51
6.2.4.	Learning analytics	53
6.3.	DISEÑO DETALLADO	56
6.3.1.	Elecciones de tecnologías y herramientas	56
6.3.2.	Diagramas detallados	66
6.3.3.	Análisis del nivel de atención	88
6.3.4.	Generación de recomendaciones de actividades	91
6.4.	IMPLEMENTACIÓN DEL PROTOTIPO	96
6.4.1.	Aplicación de captura de pantalla	96
6.4.2.	Analizador de capturas de pantalla	99
6.4.3.	Asistente del profesor	108
6.4.4.	Learning analytics	123
7.	CONCLUSIONES Y TRABAJOS FUTUROS	136
7.1.	RESULTADOS Y APRECIACIONES FINALES	136
7.2.	TRABAJOS FUTUROS	137
8.	REFERENCIAS	139
9.	ANEXOS	150
9.1.	ANEXO 1 - REPOSITORIOS DEL CÓDIGO FUENTE DEL PROTOTIPO	

LISTADO DE TABLAS

<i>Tabla 1: Comparación de trabajos revisados como parte del estado del arte. Creación propia</i>	40
<i>Tabla 2: Sistemas y aplicaciones que componen la arquitectura de la solución. Creación propia</i>	55
<i>Tabla 3: Ventajas de usar tecnologías serverless para crear sistemas. Creación propia. Información tomada de (Fee, 2020), (Red Hat, 2022), (Lassa, 2019) y (Mezzalira, Hyatt, Denti, & Jaupaj, 2022)</i>	58
<i>Tabla 4: Desventajas de usar tecnologías serverless para crear sistemas. Creación propia. Información tomada de (Fee, 2020), (Red Hat, 2022), (Lassa, 2019) y (Mezzalira, Hyatt, Denti, & Jaupaj, 2022)</i>	59
<i>Tabla 5: Elección de tecnología para ejecución de código como funciones (FaaS). Creación propia</i>	62
<i>Tabla 6:: Elección de tecnología para exponer funcionalidades a clientes (APIs). Creación propia</i>	62
<i>Tabla 7: Elección de tecnología para almacenamiento de datos en la nube. Creación propia</i>	63
<i>Tabla 8: Elección de tecnología para herramientas de Observabilidad. Creación propia</i>	63
<i>Tabla 9: Elección de tecnología para seguridad y manejo de acceso. Creación propia</i>	64
<i>Tabla 10: Elección de tecnología para comunicación basado en mensajes y entres servicios. Creación propia</i>	64
<i>Tabla 11: Elección de tecnología para herramientas de integración y despliegues. Creación propia</i>	65
<i>Tabla 12: Elección de tecnología para desarrolladores y construcción de código. Creación propia</i>	66
<i>Tabla 13: Jerarquía de diagramas dentro del diseño detallado bajo modelo C4. Creación propia</i>	68
<i>Tabla 14: Elementos dentro del diagrama de contexto bajo el modelo C4. Creación propia</i>	71
<i>Tabla 15: Elementos dentro del diagrama de contenedores del sistema "Screenshot Analyzer", o Analizador de Captura de Pantalla bajo el modelo C4. Creación propia</i>	72
<i>Tabla 16: Elementos dentro del diagrama de componentes del contenedor "Attention level analyzer", o Analizador de nivel de atención, bajo el modelo C4. Creación propia</i>	74
<i>Tabla 17: Elementos dentro del diagrama de componentes del contenedor "Screenshot app", o aplicación de captura de pantalla, bajo el modelo C4. Creación propia</i>	76
<i>Tabla 18: Elementos dentro del diagrama de contenedores "Teacher assistant", o Asistente del profesor, bajo el modelo C4. Creación propia</i>	78
<i>Tabla 19: Elementos dentro del diagrama de componentes del contenedor "Teacher feedback backend", o Backend del asistente del profesor, bajo el modelo C4. Creación propia</i>	80
<i>Tabla 20: Elementos dentro del diagrama de componentes del contenedor "Teacher feedback frontend", o interfaz de usuario del asistente del profesor, bajo el modelo C4. Creación propia</i>	82

<i>Tabla 21: Elementos dentro del diagrama de contenedores "Learning analytics", bajo el modelo C4. Creación propia</i>	<i>84</i>
<i>Tabla 22: Elementos dentro del diagrama de componentes del contenedor "Learning Analytics Backend", o Backend del sistema de Learning analytics, bajo el modelo C4. Creación propia</i>	<i>86</i>
<i>Tabla 23: Elementos dentro del diagrama de componentes del contenedor "Learning Analytics Dashboard", o Panel del sistema de Learning analytics, bajo el modelo C4. Creación propia</i>	<i>88</i>
<i>Tabla 24: Requisitos y elección de tecnologías para la aplicación de captura de pantalla. Creación propia.....</i>	<i>97</i>
<i>Tabla 25: Modelado de datos para datos guardados al iniciar la captura de pantalla. Creación propia</i>	<i>98</i>
<i>Tabla 26: Requisitos y tecnologías para analizador de capturas de pantalla. Creación propia</i>	<i>99</i>
<i>Tabla 27: Modelado de datos para el almacenamiento de resultados de análisis de nivel de atención y rostros. Creación propia.</i>	<i>105</i>
<i>Tabla 28: Requisitos y tecnologías para el "Backend" del asistente del profesor</i>	<i>108</i>
<i>Tabla 29: Modelado de datos para el asistente del profesor, usando el patrón "Single table design". Creación propia</i>	<i>113</i>
<i>Tabla 30: Requisitos y tecnologías elegidas para la implementación del módulo de learning analytics en el prototipo. Creación propia</i>	<i>123</i>
<i>Tabla 31: Modelado para la entidad de clase en la tabla "Classes". Creación propia</i>	<i>127</i>
<i>Tabla 32: Modelado para la entidad de captura de pantalla en la tabla "ClassScreenshots". Creación propia</i>	<i>128</i>
<i>Tabla 33: Modelado para la entidad de rostro en la tabla "ClassFaces". Creación propia</i>	<i>129</i>

LISTADO DE ILUSTRACIONES

<i>Ilustración 1: Comparación de cambios en el aprendizaje bajo metodología ICAP. Tomado de (Chi & Wylie, 2014)</i>	<i>13</i>
<i>Ilustración 2: Movimientos de ojo de tres sujetos. Los sujetos S1 y S2 presentaron calificaciones altas mientras que S3 presentó calificaciones bajas. Tomado de (Madsen, Júlio, Gucik, Steinberg, & Parra, 2020)</i>	<i>22</i>
<i>Ilustración 3: Precisión de los modelos evaluados para diferentes niveles de desenfoque. Tomado de (Yang, et al., 2021)</i>	<i>23</i>
<i>Ilustración 4: Vista de alto nivel del trabajo para identificar nivel de atención y relacionar con AWS Rekognition. Tomado de (Tabassum, Allen, & De, 2020)</i>	<i>24</i>
<i>Ilustración 5: Visualización de la precisión del modelo entrenado para el data set de entrenamiento y el dataset de validación. Tomado de (Tabassum, Allen, & De, 2020)</i>	<i>25</i>
<i>Ilustración 6: Componentes de learning analytics. Tomado de (Tsai, 2022)</i>	<i>27</i>
<i>Ilustración 7: Captura de pantalla de asistente de enseñanza. Tomado de (Trabelsi, Alnajjar, Ambali Parambil, Gochoo, & Ali, 2023)</i>	<i>28</i>
<i>Ilustración 8: Tipos de escalabilidad en la nube. Tomado de (Symphony Solutions, 2020).....</i>	<i>29</i>
<i>Ilustración 9: Ejemplo de una arquitectura Serverless para un servicio. Tomado de (Lefèvre, 2020)</i>	<i>30</i>
<i>Ilustración 10: Ejemplo de balanceo de carga para una aplicación de servidor. Tomado de (Key CDN, 2023).33</i>	<i>33</i>
<i>Ilustración 11: Arquitectura para una aplicación de ejemplo que usa el patrón "Fan-Out" para su comunicación asíncrona. Tomado de (Richardson & Fröhner, 2019).....</i>	<i>35</i>
<i>Ilustración 12: Diagrama de Venn sobre conceptos de aprendizaje de máquina (Machine learning) y las diferentes clases que se presentan. Tomado de (Janiesch, Zschech, & Henrich, 2021).....</i>	<i>36</i>
<i>Ilustración 13: Rendimiento de GPT-4 en pruebas académicas comparado con otros modelos similares del estado del arte. Tomado de (OpenAI, 2023)</i>	<i>37</i>
<i>Ilustración 14: Rendimiento de modelos GPT en diferentes pruebas académicas. Tomado de (OpenAI, 2023)</i>	<i>38</i>
<i>Ilustración 15: Diagrama de casos de uso de la solución. Creación propia</i>	<i>47</i>
<i>Ilustración 16: Diagrama de secuencia del sistema "Analizador de capturas de pantalla" o "Screenshot analyzer". Creación propia</i>	<i>49</i>
<i>Ilustración 17: Diagrama de secuencia del sistema "Asistente del profesor" o "Teacher assistant". Creación propia</i>	<i>51</i>
<i>Ilustración 18: Diagrama de secuencia del sistema "Learning Analytics". Creación propia.....</i>	<i>53</i>

<i>Ilustración 19: Pruebas de tiempos de arranque en frío para diferentes configuraciones. Tomado de (Cui, 2020)</i>	61
<i>Ilustración 20: Ciclo de vida de una función AWS Lambda. Tomado de (Cui, 2020)</i>	61
<i>Ilustración 21: Diagrama de contexto de la solución. Creación propia</i>	69
<i>Ilustración 22: Diagrama de contenedores para analizador de capturas de pantalla. Creación propia</i>	71
<i>Ilustración 23: Diagrama de componentes del analizador de nivel de atención. Creación propia</i>	73
<i>Ilustración 24: Diagrama de componentes para la aplicación de captura de pantalla. Creación propia</i>	75
<i>Ilustración 25: Diagrama de contenedores para el asistente del profesor. Creación propia</i>	77
<i>Ilustración 26: Diagrama de contenedores del backend para el asistente del profesor. Creación propia</i>	79
<i>Ilustración 27: Diagrama de componentes para el frontend del asistente del profesor. Creación propia</i>	81
<i>Ilustración 28: Diagrama de contenedores para el sistema de learning analytics. Creación propia</i>	83
<i>Ilustración 29: Diagrama de componentes para el backend del sistema learning analytics. Creación propia</i>	85
<i>Ilustración 30: Diagrama de componentes para el panel visual de learning analytics. Creación propia</i>	87
<i>Ilustración 31: Resultado de análisis de regresión que relaciona emociones identificadas en un rostro con nivel de atención. Tomado de (Tabassum, Allen, & De, 2020)</i>	88
<i>Ilustración 32: Recorte del código para el cálculo de nivel de atención de rostros basados en sus emociones. Creación propia</i>	90
<i>Ilustración 33: Ejemplo de ataque de “prompt injection” hacia un bot de twitter que usa el modelo GPT-3 para generar sus respuestas. Tomado de (Willison, 2022)</i>	93
<i>Ilustración 34: Ejemplo de protección contra inyección de prompt usando una verificación de entrada previa con token aleatorio en el modelo GPT-4. Creación propia</i>	93
<i>Ilustración 35: Diagrama de flujo para obtener una recomendación de actividad del modelo de procesamiento de lenguaje natural. Creación propia</i>	95
<i>Ilustración 36: Código usado para tomar capturas de pantalla en un intervalo de tiempo definido. Creación propia</i>	97
<i>Ilustración 37: Ejemplo de almacenamiento en la nube que muestra la jerarquía usada para almacenar las capturas de pantalla. Creación propia</i>	98
<i>Ilustración 38: Ejemplo de datos guardados para las diferentes clases con diferentes configuraciones. Creación propia</i>	98
<i>Ilustración 39: Diagrama de flujo de la identificación de rostros y cálculo de nivel de atención de estudiantes. Creación propia</i>	101
<i>Ilustración 40: Fragmento del código del controlador que procesa el evento de carga de la captura de pantalla. Creación propia</i>	102

<i>Ilustración 41: Fragmento del código necesario para configurar los disparos de eventos cuando se carga un nuevo archivo usando Serverless Framework. Creación propia</i>	<i>102</i>
<i>Ilustración 42: Fragmento del código que publica los resultados del análisis. Creación propia</i>	<i>103</i>
<i>Ilustración 43: Ejemplo de datos almacenados para resultados de análisis. La primera columna "classId_pk" señala la llave de partición (Partition Key), cuyo valor es el identificador de la clase (classId). En la segunda columna se encuentra "screenshot_attentionLevel_sk", que hace referencia a la llave de ordenamiento (Sort Key), la cual es una llave compuesta y separada por el carácter _. En esta llave se encuentra el identificador de la captura de pantalla (screenshotId), el nivel de atención identificado, y el índice del rostro al cual se refiere. Creación propia.</i>	<i>105</i>
<i>Ilustración 44: Ejemplo de jerarquía en el almacenamiento de imágenes de rostros en la nube. Creación propia</i>	<i>106</i>
<i>Ilustración 45: Fragmento de los permisos del rol de la función lambda. Creación propia.....</i>	<i>107</i>
<i>Ilustración 46: Fragmento de la configuración necesaria para que la función lambda generada pueda procesar mensajes de la cola. Creación propia.</i>	<i>109</i>
<i>Ilustración 47: Diagrama de flujo para procesar reporte de resultados de análisis para una captura de pantalla. Creación propia</i>	<i>110</i>
<i>Ilustración 48: Ejemplo de modelado de datos del asistente del profesor usando el patrón de "single table design". Creación propia</i>	<i>113</i>
<i>Ilustración 49: Diagrama de flujo de función "Obtener últimos rostros" del asistente del profesor. Creación propia</i>	<i>114</i>
<i>Ilustración 50: Fragmento de código del servicio de recomendación que genera una recomendación de actividad basada en el contexto de la clase. Creación propia</i>	<i>115</i>
<i>Ilustración 51: Fragmento de los permisos para el rol de la función Lambda. Creación propia.....</i>	<i>116</i>
<i>Ilustración 52: Fragmento del código del servicio de análisis que se encarga de obtener los datos del servidor. Creación propia</i>	<i>117</i>
<i>Ilustración 53: Fragmento de código del "hook" personalizado para obtener el último análisis y rostros de este. Creación propia</i>	<i>118</i>
<i>Ilustración 54: Fragmento del código de un componente de botón de React reusable. Creación propia</i>	<i>119</i>
<i>Ilustración 55: Fragmento del código que muestra el uso del componente reusable "botón" en una página. Creación propia</i>	<i>119</i>
<i>Ilustración 56: Vista inicial de la interfaz de usuario del asistente del profesor en tamaño laptop. Creación propia</i>	<i>120</i>

<i>Ilustración 57: Vista de la interfaz de usuario del asistente del profesor en tamaño laptop con el monitoreo de nivel de atención activado. Los rostros se han desenfocado para efectos de este documento. Creación propia</i>	<i>121</i>
<i>Ilustración 58: Ejemplos de actividades para "Romper el hielo" que se generan y muestran en la interfaz de usuario. Creación propia</i>	<i>122</i>
<i>Ilustración 59: Diagrama de flujo de la lógica para procesar eventos de resultados de análisis. Creación propia</i>	<i>124</i>
<i>Ilustración 60: Diagrama de entidad-relación para la base de datos de learning analytics. Creación propia</i>	<i>126</i>
<i>Ilustración 61: Estructura de las migraciones y modelos implementados usando Sequelize. Creación propia.</i>	<i>130</i>
<i>Ilustración 62: Fragmento de código para la migración de la tabla "ClassFaces". Creación propia.</i>	<i>131</i>
<i>Ilustración 63: Fragmento del código que muestra el objeto-relación para la entidad rostro de clase y la tabla "ClassFaces" usando sequelize-typescript. Se puede apreciar las relaciones con otras entidades. Creación propia.</i>	<i>132</i>
<i>Ilustración 64: Fragmento del código que muestra cómo se guarda un rostro en base de datos usando las clases del ORM. Creación propia</i>	<i>133</i>
<i>Ilustración 65: Fragmento del código para crear y configurar una instancia de base de datos AWS RDS postgres. Creación propia</i>	<i>134</i>
<i>Ilustración 66: Fragmento de la infraestructura como código para crear una VPC y configurar la base de datos dentro. Creación propia</i>	<i>135</i>

RESUMEN

En el proceso de aprendizaje, los educadores enfrentan dificultades para evaluar el nivel de atención de los estudiantes de sus clases, una métrica crucial para implementar efectivamente actividades de aprendizaje activo que mejoren los resultados del aprendizaje. Este reto se resalta aún más cuando se trata de un entorno virtual, dado que el problema se agudiza al no tener la cercanía que se tiene en un aula de clases tradicional. Este estudio aborda esta dificultad, explorando las barreras actuales en la medición de nivel de atención y propone una herramienta digital que tenga la capacidad de medir y mostrar dicho nivel de atención de manera no intrusiva. La herramienta, construida a partir de una arquitectura específicamente diseñada, tiene la capacidad de capturar, almacenar y mostrar de manera visual el nivel de atención de cada estudiante que participa de una clase virtual. Se construyó un prototipo a partir de esta arquitectura que además integra un servicio de Procesamiento de Lenguaje Natural, específicamente utilizando el modelo de lenguaje GPT-4, para sugerir actividades de aprendizaje activo al docente. Como trabajo futuro, se propone expandir el prototipo para que pueda ser de utilidad para los estudiantes, así como verificar su utilidad en otros contextos como conferencias o charlas.

Palabras Claves: Medición de Nivel de Atención, Learning Analytics, Asistente Virtual de Enseñanza, Aprendizaje Virtual, Aprendizaje Activo, GPT-4, Inteligencia Artificial en Educación.

ABSTRACT

In the learning process, educators face challenges in assessing the level of attention of their students in classes, a crucial metric for effectively implementing active learning activities that improve learning outcomes. This challenge is even more pronounced in a virtual environment, as the issue is exacerbated by the lack of proximity found in a traditional classroom. This study addresses this difficulty, exploring the current barriers in measuring attention levels and proposing a digital tool capable of non-intrusively measuring and displaying this level of attention. The tool, built from a specifically designed architecture, has the ability to capture, store, and visually display the attention level of each student participating in a virtual class. A prototype was built from this architecture, which also integrates a Natural Language Processing service, specifically using the GPT-4 language model, to suggest active learning activities to the teacher. For future work, it is proposed to expand the prototype so that it can be useful for students, as well as to verify its usefulness in other contexts such as conferences or talks.

Keywords: Attention Level Measurement, Learning Analytics, Virtual Teaching Assistant, Virtual Learning, Active Learning, GPT-4, Artificial Intelligence in Education.

1. INTRODUCCIÓN

En esta sección se presenta una introducción acerca del problema que se aborda en este trabajo y propone una herramienta tecnológica innovadora para solucionarlo. Se examinan los desafíos asociados con el aprendizaje en línea y que funcionalidades debe tener la herramienta propuesta.

El mundo académico, así como muchos sectores de la sociedad, resultó impactado de manera significativa cuando la pandemia causada por el COVID-19 hizo que las instituciones educativas adaptaran rápidamente una transición hacia el aprendizaje en línea y las clases virtuales (Crawford, Jurgen, Butler-Henderson, & Malkawi, 2020). Si bien, esto permitió que se continuara con los currículos y planes educativos actuales, también generó desafíos únicos para los educadores, los cuales debieron adaptar sus métodos de enseñanza a un nuevo entorno virtual. Uno de estos principales retos es la adaptación de técnicas de aprendizaje activo para mejorar los resultados de enseñanza de los estudiantes. Si bien, en la presencialidad también se tiene este reto, en la virtualidad se amplifica esta dificultad ya que los educadores no pueden determinar con facilidad si sus estudiantes están prestando atención o no.

Un estudiante que haga parte de una clase se beneficia más cuando se implementan técnicas de aprendizaje activo en vez de solo escuchar o ver información. Se llama aprendizaje activo cuando los estudiantes piensan y usan la información de la clase de una manera significativa. En concreto, cuando los estudiantes se involucran con la información: analizando, juntando y evaluando lo que han aprendido. Se desarrolló y estudió un marco de trabajo basado en el aprendizaje activo que comienza como aprendizaje “pasivo”, para luego convertirse en “activo”, después en “constructivo” y, finalmente, en “interactivo”. En el experimento que se realizó para validar este marco de trabajo se encontró que en cada etapa el aprendizaje aumentaba entre un 8% y 10% (Chi & Wylie, 2014).

CATEGORY <i>Characteristic</i>	PASSIVE <i>Receiving</i>	ACTIVE <i>Manipulating</i>	CONSTRUCTIVE <i>Generating</i>
Example activities	Listening to explanations; Watching a video	Taking verbatim notes; Highlighting sentences	Self-explaining; Comparing and contrasting
Knowledge-change processes	Isolated “storing” processes in which information is stored episodically in encapsulated form without embedding it in a relevant schema, no integration	“Integrating” processes in which the selected & emphasized information activates prior knowledge & schema, & new information can be assimilated into the activated schema.	“Inferring” processes include: integrating new information with prior knowledge; inferring new knowledge; connecting, comparing & contrasting different pieces of new information to infer new knowledge; analogizing, generalizing, reflecting on conditions of a procedure, explaining why something works.
Expected changes in knowledge	New knowledge is stored, but stored in an encapsulated way.	Existing schema is more complete, coherent, salient, and strengthened.	New inferences create new knowledge beyond what was encoded, thus existing schema may become more enriched; procedures may be elaborated with meaning, rationale and justifications; and mental models may be accommodated; and schema may be linked with other schemas.

Ilustración 1: Comparación de cambios en el aprendizaje bajo metodología ICAP. Tomado de (Chi & Wylie, 2014)

En un estudio realizado en 2011, Se usó una encuesta para averiguar la percepción de estudiantes sobre el valor y relevancia del aprendizaje activo y las actividades colaborativas durante sus clases. De los 94 estudiantes, 91 dijeron que las actividades de aprendizaje activo ayudaron a que entendieran y aprendieran los conceptos de manera más profunda. La razón más común fue que aprender activamente les permitió ser partícipes activos en su propio aprendizaje. Las discusiones en clase o en grupos pequeños les permitió formar su propia opinión y cuestionar las ideas presentadas por el profesor o expresadas por sus compañeros. Las actividades de aprendizaje activo ayudaron a mantener el interés y la atención durante las clases. La variedad de las actividades y el equilibrio entre participación de estudiantes y explicaciones del profesor resultaron cruciales, ya que ayudaron a mantener la concentración y permitieron que los estudiantes estuvieran más alertas y, por ende, estuvieran menos distraídos. El uso de ejemplos prácticos y realistas motivó a estudiantes y mantuvo su interés (Cavanagh, 2011).

Los educadores más efectivos son aquellos que aseguran el aprendizaje de sus estudiantes y que practican el conocimiento a través de actividades, pero siempre después de haber dado el material básico del tema de la clase. Algunas técnicas que usaron incluyen:

- Comenzar la clase con un resumen corto de lo aprendido en sesiones pasadas.
- Pedir a los estudiantes que proporcionen respuestas a preguntas a un compañero, o que elaboren un resumen en un papel y lo compartan con otro estudiante.
- Pedir a los estudiantes que expliquen lo que han aprendido.
- Pensar en voz alta y demostrar los pasos del pensamiento de manera ordenada.

Además, los profesores más efectivos dedican mayor tiempo a la práctica guiada, haciendo preguntas y corrigiendo errores. Pedir y supervisar la práctica autónoma es otra táctica necesaria para que las habilidades y el conocimiento de un estudiante se convierta en automático (Rosenshine, 2012).

En este trabajo se aborda la necesidad de identificar el nivel de atención en estudiantes que participan de una clase virtual, de manera no intrusiva, para facilitar al profesor aplicar métodos de aprendizaje activo en su proceso de enseñanza. Para esto, se propone una herramienta tecnológica que el educador pueda usar para visualizar el nivel de atención de los estudiantes a medida que transcurre la clase. También, puede usar esta información del nivel de atención para realizar actividades de aprendizaje activo, para lo que se propone que la herramienta genere una recomendación de actividad haciendo uso de un modelo de procesamiento de lenguaje natural (GPT-4) para generar actividades que tengan relevancia para el contexto de la clase actual. Por último, la herramienta tecnológica pueda capturar datos históricos y visualizarlos a modo de información adicional para el educador, de manera que los profesores puedan hacer una revisión de la clase y de los niveles de atención medidos por clase.

Para la construcción se hizo un análisis detallado de lo que se requería cumplir con la herramienta propuesta. Siguiendo patrones de desarrollo de software comunes como la definición de requisitos, descripción de historias de usuario, y señalando consideraciones y suposiciones, se da un diseño de arquitectura para una herramienta tecnológica que pueda abordar los desafíos expuestos. También se construye un prototipo basado en dicha arquitectura, mostrando un diseño detallado y elección de tecnologías acorde a la arquitectura.

En título del trabajo, “VTA/LRS: Una aproximación a la implementación de un asistente virtual como herramienta de soporte al proceso de enseñanza”, las siglas “VTA/LRS” hacen referencia a la funcionalidad del prototipo, por sus siglas en inglés: “**V**irtual **T**eaching **A**ssistant **L**earning **R**ecommendation **S**ystem”.

El resto de este documento se estructura de la siguiente manera: Primero se presenta un planteamiento del problema donde se expande la problemática que aborda el trabajo, se define la pregunta y la idea de investigación. En la sección siguiente se da una justificación del trabajo, se explica su relevancia y contexto. Luego se presenta la sección de objetivos, donde se afianzan los objetivos que

quiere cumplir el trabajo. Se da un contexto de diferentes temas relevantes al trabajo a través del marco teórico o marco conceptual. Se presenta el diseño metodológico usado para el trabajo. Luego se aborda el desarrollo del trabajo, donde se explica con detalle los pasos necesarios para diseñar y construir el prototipo mencionado. Al final, se presenta una sección de conclusiones acerca del trabajo desarrollado.

2. PLANTEAMIENTO DEL PROBLEMA

Para los estudiantes el nivel de atención y compromiso que demuestran con la clase impacta su proceso de aprendizaje de manera directa. En estudios de diferentes universidades de estados unidos se encontró que el 40% de los estudiantes experimentan aburrimiento de manera frecuente y menos del 20% de los estudiantes hacen preguntas debido a baja participación en la clase. Se encuentra que los estudiantes que estén comprometidos y participen de manera activa en una clase tienen mayor motivación y reciben mayor participación del educador. Por el contrario, los estudiantes desconectados con la clase encuentran mayores dificultades como consecuencia de esta desconexión (Subramainan & Mahmoud, 2020).

Cuando el educador no encuentra recursos, ideas o materiales para manejar los retos sociales de la clase, sus estudiantes muestran poco interés y rendimiento en sus tareas asignadas. Es por esto que se sugieren diferentes técnicas de interacción para recuperar el interés de los estudiantes. En particular, se identifican los siguientes:

- Estimular discusión, debates, exploración y descubrimiento.
- Proveer espacios para sesiones de preguntas/respuestas.
- Proveer retroalimentación.
- Pedir opinión de estudiantes y tomar acción sobre la misma.
- Quices en línea como herramientas de autoevaluación.
- Reducir tiempo de explicación para incrementar tiempo de discusión grupal.
- Dar ejemplos y ejercicios aplicados a la vida real.
- Usar herramientas multimedia como apoyo a la educación.
- Actividades de autoevaluación y autorreflexión.

(Subramainan & Mahmoud, 2020)

Las personas pueden ser activas y mantenerse unidas las una a las otras, o pueden tener comportamientos más pasivos y solitarios. Cuando estamos en nuestro mejor momento, los seres humanos somos curiosos y nos movemos por nuestra propia iniciativa. De la misma manera, las personas que trabajan duro y muestran energía para hacer las cosas parecen ser más especiales y muestran una actitud buena y constante. Sin embargo, a veces podemos sentirnos agobiados, lo que nos lleva a

evitar el crecimiento o la responsabilidad, sin importar su estado social u origen cultural (Ryan & Deci, 2000).

Cuando la pandemia ocasionada por el COVID-19 comenzó, muchas instituciones educativas tuvieron que adaptarse y cambiar su pedagogía en ciclos rápidos. Obligó a cambiar la educación tradicional presencial por métodos virtuales síncronos y asíncronos. Los educadores encontraron un reto en mantener a sus estudiantes activos y por tanto se sugieren varias alternativas para mantener la atención de sus estudiantes y ayudar a los estudiantes a participar más. Algunas de esas técnicas son:

- Documentación en línea: Ayuda a que los estudiantes trabajen en equipo y entiendan los conceptos de manera conjunta. Notas de clase, foros de discusión, presentaciones, entre otros.
- Interacciones estudiante-estudiante: Los estudiantes expertos juegan un papel clave en las clases en línea, ayudando a facilitar la instrucción de forma directa o indirecta. Monitoreando preguntas en chat, promoviendo la discusión en grupos o asistiendo con problemas tecnológicos.
- Maneras de comunicación informales: Un reto que se evidencia con más amplitud en clases virtuales que en clases presenciales es el de fomentar la participación estudiantil. Los estudiantes en clases en línea tienden a ser más tímidos a la hora de compartir sus ideas o hacer preguntas de manera verbal, sin embargo, se muestran dispuestos al usar el chat o “emojis” ya que están más familiarizados con esos métodos de comunicación.
- Salas digitales de trabajo grupal: Estas son salas virtuales que sirven como ayuda para el aprendizaje colaborativo de los estudiantes. Ofrecen la oportunidad de interactuar y trabajar en conjunto. Se pueden usar al inicio de la clase para estimular conocimiento previo, a mitad de clase para asentir los conocimientos, o al final para consolidar ideas y conclusiones. Estas salas ofrecen ayudas digitales útiles para los estudiantes como pizarras digitales, chat, compartir pantalla, entre otros.

Al usar y adaptarse a estas técnicas de enseñanza, se puede aumentar la participación estudiantil en un entorno de clase virtual. Al usar estos enfoques para la enseñanza virtual ayuda a los estudiantes que tienen dificultades para hablar en público, tienen discapacidades o su principal idioma no es el de la clase y que sin estas ayudas tendrían dificultades para su proceso de aprendizaje virtual (Gamrat, Lenze, Bardzell, & Glantz, 2021).

Una buena educación trasciende el medio por el que es impartido, ya sea en un aula virtual o de manera presencial. Lo que funciona bien en la pedagogía cara a cara puede también funcionar en ambientes en línea. Pero los cursos en línea tienen problemas que no se dan en cursos presenciales. Algunos de estos problemas son que los estudiantes muestran falta de motivación y falta de soporte. Esto se ve reflejado en índices de abandono de entre 10-20% en cursos en línea con respecto a cursos presenciales. En el aprendizaje en línea, los estudiantes necesitan más

que solo recibir la información del curso; necesitan tener motivación y ganas de aprender, así como recibir ayuda. El profesor debe tener buena presencia, dar instrucciones claras, que el material sea útil y que las tareas hagan que los estudiantes se vuelvan participantes activos de su propio aprendizaje, son cosas importantes para ayudar a los estudiantes a terminar sus cursos con éxito. Para ser tan buenos como un curso presencial, un curso virtual debería (Nilson & Goodson, 2017).

- Fomentar el diálogo con los estudiantes lo más posible.
- Crear oportunidades para que los estudiantes trabajen en grupos.
- Usar técnicas de aprendizaje activo.
- Ayudar a los estudiantes asignando actividades y tareas que fomenten el autoaprendizaje.
- Asignar desafíos y actividades retadoras, premiando a los estudiantes cuando entienden el porqué de estos retos y su importancia en la vida real.

(Nilson & Goodson, 2017)

El estudio a distancia nos permite aprender desde cualquier lugar, facilitándonos el acceso al conocimiento. Sin embargo, este tipo de aprendizaje también trae sus propios desafíos. A veces, aprender en línea puede hacernos sentir solos y esta soledad puede dificultar nuestro aprendizaje (Borup, West, & Graham, 2012).

Un curso en internet puede ser tan bueno como uno en persona. Para alcanzar ese nivel, los cursos en línea deben usar maneras de aprender que sean basadas en el aprendizaje activo. Esto incluye aspectos cooperativos y colaborativos, a través de actividades que fomenten el trabajo en conjunto y colaboración. Además, el profesor debe tener una presencia significativa. Se ha encontrado que los estudiantes de cursos en línea pueden tener un mejor proceso de aprendizaje que los estudiantes de cursos presenciales en algunos casos. Para que los cursos virtuales sean realmente efectivos es muy importante que la forma de enseñar sea buena y que se generen oportunidades para hablar entre los estudiantes y con el profesor. Las actividades colaborativas se han estudiado, y se ha encontrado que las mejoras son las que hacen que los estudiantes colaboren entre sí y que trabajen juntos, impulsando un pensamiento crítico a través de preguntas e interacciones. Un debate bien hecho puede facilitar la cooperación y convertirse en una parte importante de la pedagogía (Dixson, 2010).

El aprendizaje activo es un enfoque pedagógico que hace que los estudiantes se vuelvan participantes activos de su propio proceso de aprendizaje. Los estudiantes más participativos recuerdan y retienen información por mayor tiempo. Un problema común en las clases es la falta de atención por parte de los estudiantes. Este problema puede reducirse si el educador implementa técnicas de aprendizaje activo. Como parte de un estudio se impartieron clases sin el uso de estas técnicas de aprendizaje activo. Los resultados muestran que los estudiantes de las clases que no usan estas técnicas comienzan a disminuir su atención a los 15 minutos de clase. Conforme avanza la clase, su nivel de atención sigue disminuyendo hasta alcanzar

niveles extremos en donde los estudiantes se dedican a otras actividades o, incluso, a dormir. Un seguimiento que se hizo 24 horas después reveló que los estudiantes solo recordaban pocos detalles, y en muchos casos eran detalles incorrectos (Bonwell & Eison, 1991).

A pesar de los avances tecnológicos en la educación en línea, la participación y el compromiso de los estudiantes siguen siendo desafíos importantes que afectan los resultados del aprendizaje. Si bien el aprendizaje activo ha demostrado ser un método efectivo para mejorar estos resultados, su implementación en un entorno de aula virtual presenta retos adicionales. Uno de los principales desafíos es la falta de retroalimentación no verbal de los estudiantes sobre su nivel de atención, lo que dificulta la adaptación y aplicación de técnicas de aprendizaje activo en tiempo real.

Por lo tanto, es esencial explorar cómo determinar y monitorear el nivel de atención de los estudiantes en un aula virtual.

2.1. PREGUNTA DE INVESTIGACIÓN

¿Cómo construir una herramienta tecnológica que permita determinar, de manera no intrusiva, el nivel de atención de estudiantes universitarios en una clase virtual y sugiera al educador actividades de aprendizaje activo?

2.2. IDEA DE INVESTIGACIÓN

Construir un prototipo de herramienta tecnológica que permita medir, de manera no intrusiva, el nivel de atención de estudiantes universitarios en una clase virtual y sugiera al educador actividades de aprendizaje activo.

3. JUSTIFICACIÓN

La educación en línea se ha convertido en un elemento importante para garantizar la enseñanza sin importar distancia. Sin embargo, a pesar de los avances tecnológicos, aún se enfrentan desafíos considerables para garantizar una participación activa y compromiso estudiantil óptimos. En este contexto, identificar el nivel de atención de los estudiantes, de manera no intrusiva, en un aula virtual se ha convertido en un obstáculo significativo.

Dicho nivel de atención de los estudiantes proporciona a los docentes pistas cruciales y comprensión del proceso real de aprendizaje, permitiendo actuar oportunamente e implementar técnicas pedagógicas como las de aprendizaje activo, las cuales han demostrado ser efectivas para mejorar el aprendizaje y la retención de la información por parte de los estudiantes.

Este trabajo de tesis es de importancia, ya que busca explorar formas de monitorear el nivel de atención de estudiantes en un aula virtual de manera no intrusiva. El objetivo es desarrollar una herramienta tecnológica que facilite el identificar el nivel de atención de los estudiantes y, de esta manera, facilite la implementación de técnicas de aprendizaje activo en entornos virtuales para que un educador pueda apoyarse en estas mediciones y ajustar su pedagogía para lograr una mejor enseñanza.

El impacto de este trabajo tiene el potencial de ser significativo y de largo alcance. Si se cumplen los objetivos propuestos, se podría mejorar la calidad de la educación en línea, impulsando la participación de los estudiantes y mejorando sus resultados. Los resultados del trabajo podrían ayudar a implementar una pedagogía apoyada por una herramienta tecnológica para la enseñanza en línea, lo cual es muy relevante para la sociedad actual.

Este trabajo tiene como objetivo contribuir a la ciencia de la educación y a la tecnología educativa en entornos virtuales, entendiendo que en trabajos futuros se puede explorar su aplicación en entornos presenciales e incluso en otras áreas y contextos.

Por último, este trabajo consiste en el entendimiento sobre la optimización de la aplicación de técnicas de aprendizaje activo en un entorno en línea. Al intentar mejorar el conocimiento en este tema, este trabajo tiene el objetivo de generar un impacto valioso y significativo en la evolución de las estrategias de enseñanza apoyadas en herramientas tecnológicas en entornos virtuales.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Construir un prototipo de herramienta tecnológica capaz de medir el nivel de atención de estudiantes, de manera no intrusiva, en una clase virtual para facilitar el uso de técnicas de aprendizaje activo por parte del instructor.

4.2 OBJETIVOS ESPECÍFICOS

- Describir los desafíos actuales que enfrentan los docentes en la medición no intrusiva del nivel de atención de los estudiantes en una clase virtual y la implementación de técnicas de aprendizaje activo que ayuden a mejorar dichos niveles de atención.
- Diseñar la arquitectura de una herramienta tecnológica que permita determinar, de manera no intrusiva, el nivel de atención de los estudiantes en un entorno de aula virtual y facilite el uso de técnicas de aprendizaje activo por parte del instructor.
- Implementar un prototipo de herramienta tecnológica a partir de la arquitectura descrita.

5. MARCO TEÓRICO Y ESTADO DEL ARTE

En esta sección se presentan diferentes temas relacionados con este trabajo, en particular se exploran otros trabajos que abordan técnicas para medir el nivel de atención en estudiantes y analizar avances en el aprendizaje en la educación en línea. Se revisan trabajos sobre asistentes virtuales para la enseñanza, que relevancia tiene con el trabajo presentado en este documento y que resultados han alcanzado. También se hace un análisis de las diferentes arquitecturas de software y una revisión sobre avances en modelos generativos de lenguaje.

5.1. MARCO TEÓRICO Y CONCEPTUAL

Durante la interrupción de la educación presencial tradicional a causa del COVID-19, los profesores tuvieron problemas para adaptarse a la nueva normalidad de clases en línea. Aunque la investigación sobre el aprendizaje en línea es limitada, existen estrategias que se enfocan en la atención, compromiso y participación activa de los estudiantes, las cuales se muestran a través de la participación activa y el esfuerzo por entender temas difíciles. Es importante que los estudiantes se sientan respetados y valorados para promover su compromiso y participación, y así lograr los objetivos de enseñanza de la clase (Coalition for Psychology in Schools and Education, 2020).

5.1.1. IDENTIFICAR ATENCIÓN USANDO IMÁGENES

En un estudio realizado en 2020, investigadores de la universidad de nueva york hicieron un estudio sobre cómo medir el nivel de atención en estudiantes mientras estos estudiantes observan un video instruccional. En el estudio midieron la atención en los videos en línea rastreando los movimientos oculares de los estudiantes. En el experimento, se mostró el mismo video a dos grupos de estudiantes: un grupo sin ninguna distracción y al otro se le pidió hacer una operación aritmética mental, que los mantenía distraídos. El estudio reveló que los estudiantes atentos tienen movimientos oculares similares al ver los videos y que la sincronización de movimientos oculares puede usarse para predecir el rendimiento individual de aprendizaje. Finalmente, concluye que la educación en línea puede adaptarse en tiempo real al nivel de atención de estudiantes y su aplicación podría ir más allá del contexto educativo, pudiendo aplicarse inclusive al entretenimiento, publicidad o política (Madsen, Júlio, Gucik, Steinberg, & Parra, 2020).

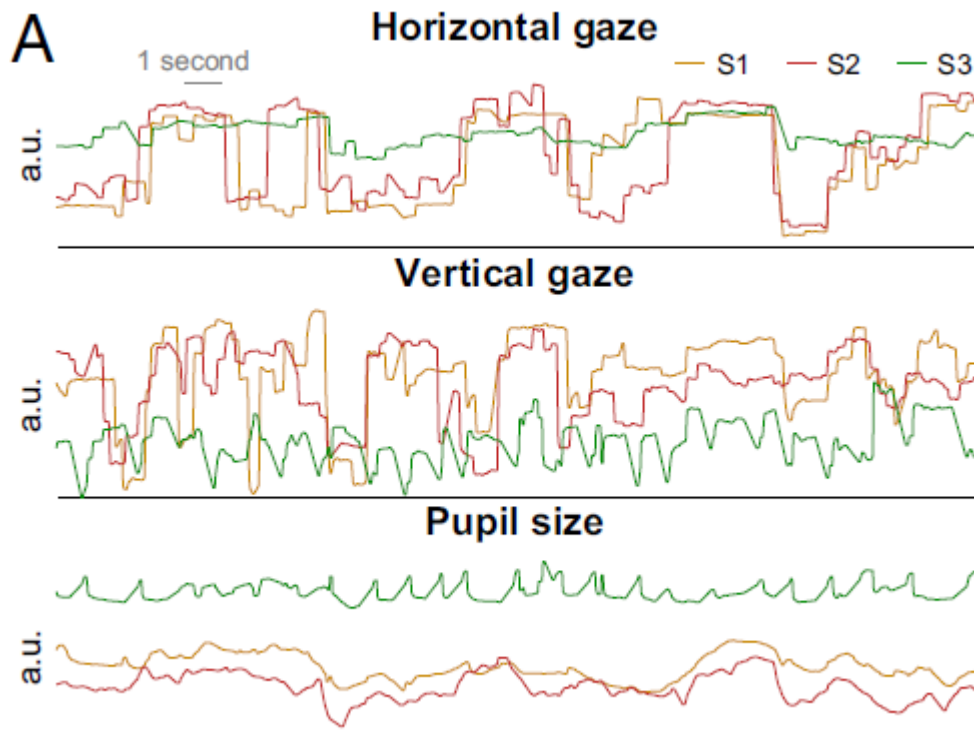


Ilustración 2: Movimientos de ojo de tres sujetos. Los sujetos S1 y S2 presentaron calificaciones altas mientras que S3 presentó calificaciones bajas. Tomado de (Madsen, Júlio, Gucik, Steinberg, & Parra, 2020)

Este estudio es un buen indicador de nivel de atención, sin embargo, requiere precisión en la captura de movimiento ocular por parte de los estudiantes, lo que implica que cada estudiante debe estar en una posición específica y mirando la pantalla para que la predicción sea efectiva.

Un estudio publicado en 2021 se enfoca en abordar las limitantes de las plataformas actuales de aprendizaje en línea en cuanto al monitoreo de los comportamientos de aprendizaje y evaluación de los mismos en la enseñanza. Los investigadores proponen analizar comportamientos de aprendizaje basándose en reconocimiento de emociones a través de imágenes. Mediante técnicas de transformación, extraen cuadros clave de imágenes de rostros humanos. Establecen un modelo de clasificación de emociones de aprendizaje basado en expresiones faciales y relacionan el modelo de clasificación de emociones para el aprendizaje en línea basado en la atención. Los resultados demuestran que el algoritmo propuesto es eficaz para analizar los comportamientos de aprendizaje a través de reconocimiento de emociones en imágenes (Wang, 2021).

Este estudio demuestra que es posible realizar un análisis de emociones en imágenes y correlacionar con el nivel de atención de los estudiantes usando técnicas de aprendizaje de máquina.

En 2020, se realizó un estudio para evaluar y comparar los 5 sistemas para reconocer imágenes de fotos de rostros para el reconocimiento de emociones faciales. Los sistemas probados fueron AWS Rekognition, Baidu, Face++, Azure y Affectiva. Las emociones que se evaluaron fueron sorpresa, miedo, disgusto, felicidad, tristeza, enojo y neutro. En primer lugar, se usaron imágenes limpias, iluminadas y que no tenía ruido, distorsión o rotación. Luego, se cambiaron las imágenes para añadirles características adicionales, una a la vez: rotación, corte parcial de la imagen, y variaciones de brillo, borrosidad y ruido. Los resultados del estudio indican que diferentes sistemas son más efectivos para identificar distintas emociones, debido a que cada sistema fue entrenado con un dataset propio y por lo tanto tiene cierto sesgo. El estudio concluye que es importante identificar qué tipo de cambio hay en una imagen y usar el sistema que mejor pueda manejar ese cambio. (Yang, et al., 2021)

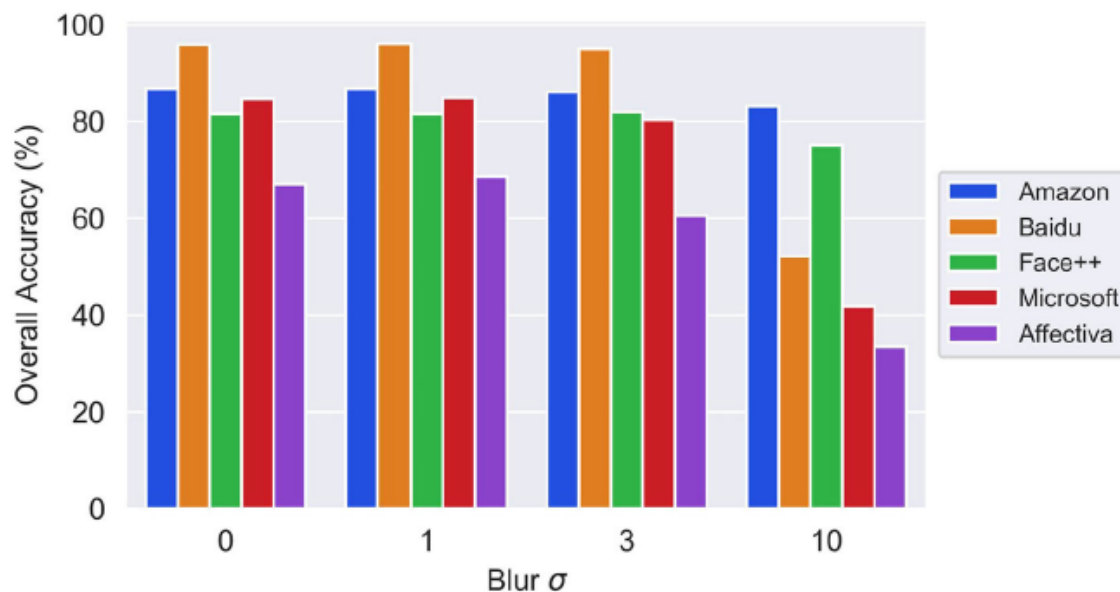


Ilustración 3: Precisión de los modelos evaluados para diferentes niveles de desenfoque. Tomado de (Yang, et al., 2021)

En un estudio realizado en 2020, un grupo de investigadores de Estados Unidos realizó dos pruebas para entender cuanta atención prestan los estudiantes en una clase. Para ello, crearon un dataset y un modelo propio utilizando imágenes de estudiantes mientras estos recibían una clase.

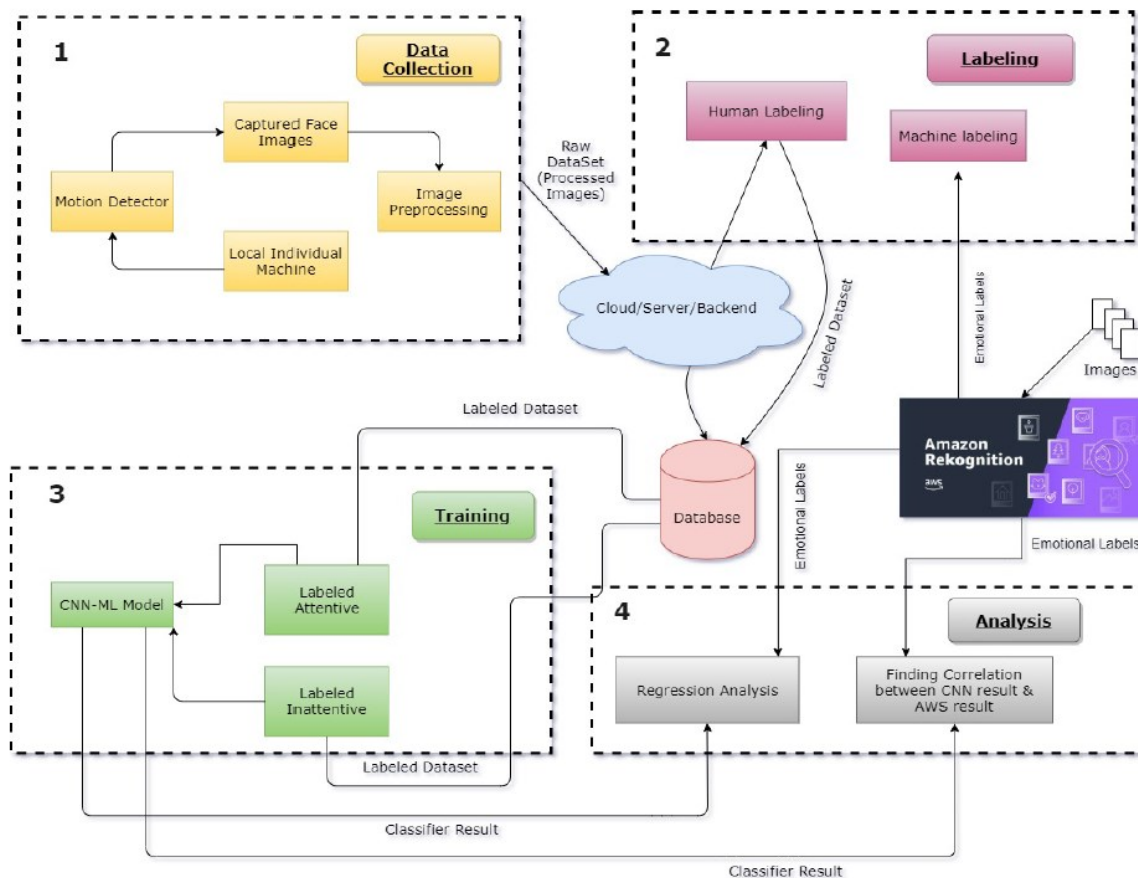


Ilustración 4: Vista de alto nivel del trabajo para identificar nivel de atención y relacionar con AWS Rekognition. Tomado de (Tabassum, Allen, & De, 2020)

Para la recolección de datos, tomaron videos de los estudiantes y, en particular, de sus rostros. Dichos videos fueron usados para extraer imágenes de los rostros de los estudiantes. Luego, cada imagen tomada pasó por un proceso de etiquetamiento en el cual usaron utilizaron criterios específicos para determinar si el estudiante estaba prestando atención o no basado en la imagen de su rostro. Esto les permitió entrenar un modelo de aprendizaje de máquina para determinar el nivel de atención de un estudiante basado en la imagen de su rostro.

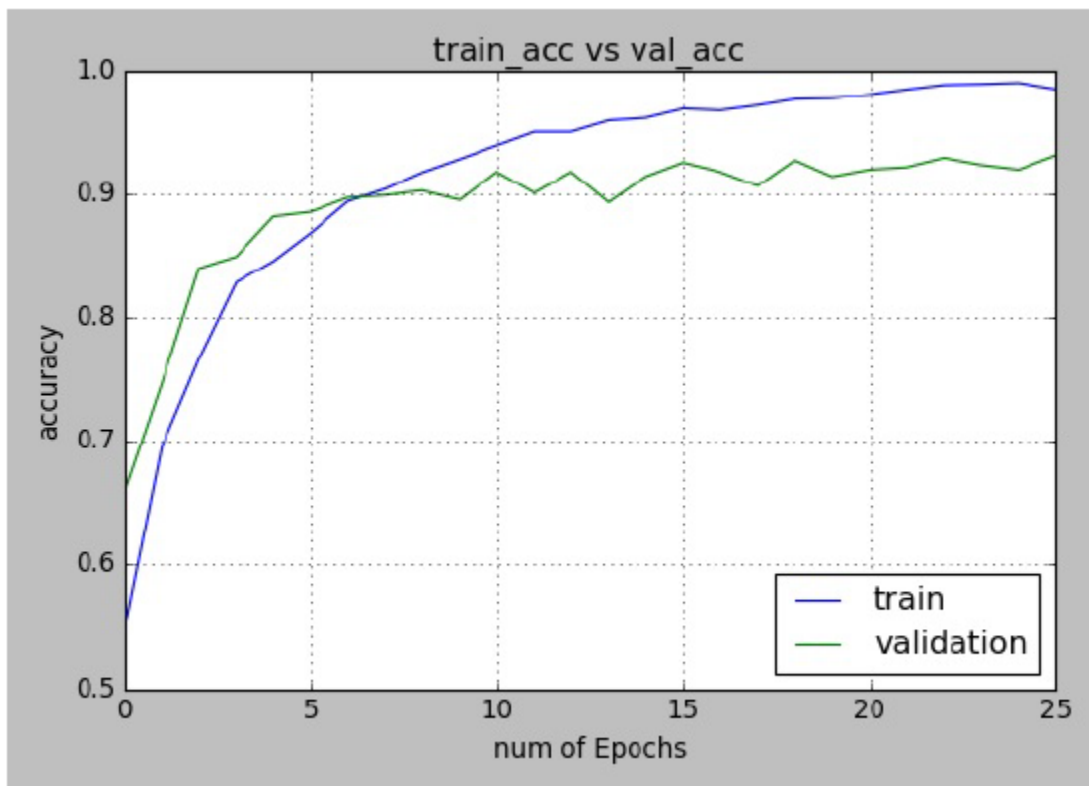


Ilustración 5: Visualización de la precisión del modelo entrenado para el data set de entrenamiento y el dataset de validación. Tomado de (Tabassum, Allen, & De, 2020)

Se usó un análisis estadístico con la ayuda de “AWS Rekognition” para encontrar la relación entre las emociones identificadas en el rostro de una persona y su nivel de atención encontrado en el dataset y modelo.

Usando técnica de correlación se encontró que cuando un estudiante es catalogado como tranquilo, confundido, asustado o sorprendido, es probable que esté prestando atención, mientras que un estudiante catalogado como feliz o triste es afectado de manera negativa. El enojo se descartó ya que no parecía útil para la prueba. Usando la técnica de mínimo cuadrado ordinario (OLS) se pudo estimar el nivel de atención de un rostro a partir de sus emociones clasificadas por “AWS Rekognition”.

El estudio concluye que las expresiones faciales de un estudiante son un indicador de su nivel de atención en una clase, y podría ser útil en el ámbito. Sugieren que un sistema de monitoreo de atención de estudiantes, o del promedio de la clase en un período de tiempo, podría ser útil como apoyo al proceso pedagógico. También concluye que esta metodología podría extenderse a otros sectores además del educativo (Tabassum, Allen, & De, 2020).

5.1.2. LEARNING ANALYTICS

SOLAR se enfoca en la medición, recolección, análisis y reporte de datos relacionados con los alumnos y sus contextos, con el fin de comprender y mejorar el aprendizaje y el ambiente en el que se produce. El campo de Learning Analytics, emergente tanto en el ámbito académico como en el comercial en la última década, aborda tres temas principales de investigación:

1. Aprendizaje, que incluye investigación educativa, ciencias del aprendizaje y evaluación, y tecnología de la educación.
2. Analítica, que comprende estadística, visualización, ciencias de la computación y de los datos, e inteligencia artificial.
3. Diseño centrado en el ser humano, enfocado en usabilidad, diseño participativo y pensamiento de sistemas sociotécnicos.

Por mucho tiempo hemos estudiado cómo mejorar los procesos de enseñanza y aprendizaje, se ha analizado los datos de escuelas y universidades con el objetivo de mejorar este proceso de enseñanza. Learning Analytics se basa en estos análisis y aprovecha las nuevas oportunidades que ofrecen los datos digitales actuales, así como las técnicas de análisis computacional proporcionadas por la ciencia de los datos y la inteligencia artificial.

Los principales usos de Learning Analytics incluyen la predicción del éxito académico, la identificación de estudiantes en riesgo, el apoyo al desarrollo de técnicas de aprendizaje de los estudiantes, la entrega de retroalimentación personalizada y oportuna, y el fomento de habilidades importantes como colaboración, pensamiento crítico, comunicación y creatividad.

En cuanto a metodologías, se destacan la Analítica Descriptiva, Diagnóstica, Predictiva y Prescriptiva, las cuales aportan conocimientos sobre el pasado, explican por qué ocurrieron ciertos eventos, anticipan el futuro y aconsejan sobre posibles resultados, respectivamente. Además, se utiliza la visualización de datos para proporcionar métricas claras a nivel de programa/grado sobre los estudiantes, su inscripción, resultados, etapa del programa y retroalimentación.

En resumen, Learning Analytics representa un ciclo continuo que involucra a los alumnos, los datos, las decisiones e intervenciones. La información obtenida de los alumnos permite mejorar la enseñanza e informar decisiones en las prácticas educativas (Tsai, 2022).

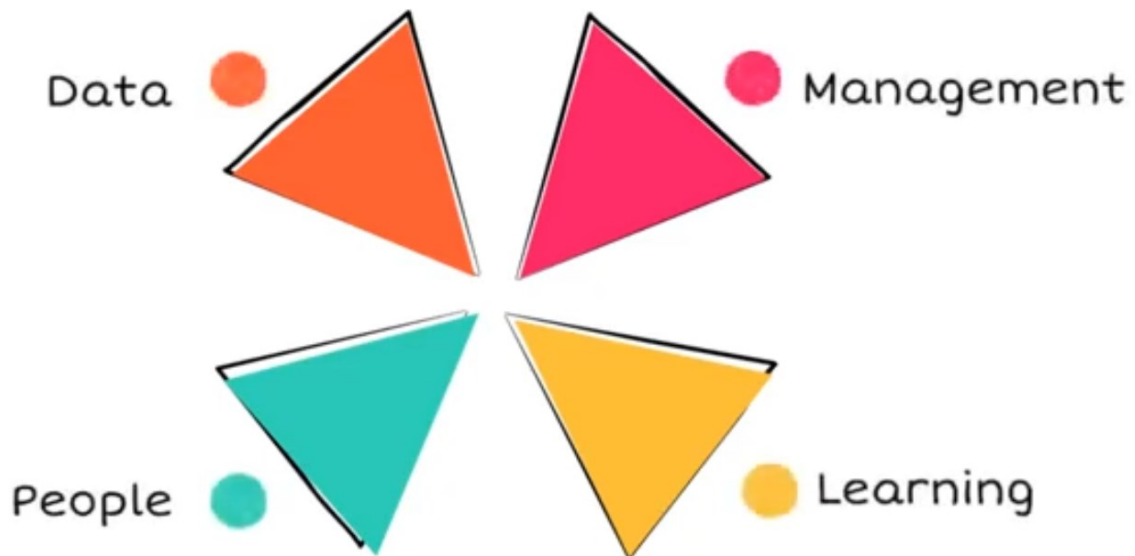


Ilustración 6: Componentes de learning analytics. Tomado de (Tsai, 2022)

Una revisión de literatura realizada en 2019 sobre learning analytics se centra en cómo utilizar los análisis del aprendizaje para obtener información útil que lleve a tomar mejores decisiones sobre el proceso pedagógico en un contexto educativo. El estudio hace énfasis también en la necesidad de un análisis de aprendizaje que pueda ayudar a inducir un cambio en el comportamiento del usuario. Sin embargo, se menciona dependencia de datos digitales y falta de estos, lo cual limita e impide tener una visión completa de cómo se da el aprendizaje.

El estudio sugiere usar más medidas complementarias que incorporen aspectos de los estudiantes, y destaca tener en cuenta el contexto de aprendizaje. Además, se debería validar cómo usar datos para tomar decisiones sobre el proceso de aprendizaje y la pedagogía (Mangaroska & Giannakos, 2019).

5.1.3. ASISTENTES DE ENSEÑANZA

Se ha estudiado como un asistente virtual en forma de “chatbot” puede apoyar el proceso de aprendizaje de estudiantes. Los chatbots son robots que tienen la capacidad de interactuar con las personas usando texto o voz. Los chatbots son buenos para responder preguntas o dar información oportuna. Se ha demostrado que el uso de asistentes virtuales para estudiantes como un chatbot ayuda a los estudiantes a responder preguntas en horas donde el profesor no está disponible, además existe un desafío en las interacciones estudiante-profesor, ya que los estudiantes pueden evitar hacer preguntas por miedo a retroalimentación negativa por parte del profesor si la pregunta se considera “básica”. En general, los chatbots

son un asistente virtual de enseñanza enfocada a los estudiantes que puede usarse como una herramienta útil para mejorar el aprendizaje adaptativo y autoeficacia de los estudiantes (Essel, Vlachopoulos, Tachie-Menson, Johnson, & Baah, 2022).

En un estudio publicado en 2023, investigadores usaron técnicas de aprendizaje profundo y aprendizaje de máquina para detectar y reconocer comportamientos de estudiantes en tiempo real. Encontraron que los modelos más robustos eran mejores para encontrar objetos pequeños con mayor precisión, pero necesitaban mayores recursos computacionales y de energía. La herramienta logró identificar diferentes acciones o comportamientos de los estudiantes como: Concentrado, Alzando la mano, Leyendo un libro, Aburrido, Usando su teléfono, entre otras. Al implementar la técnica de medición en un aula de clase se recopilaban datos que proporcionaron información valiosa sobre el nivel de atención y las emociones de los estudiantes, permitiendo al profesor tener una visión mejor sobre la atención de sus estudiantes y ayudando a mejorar el proceso de aprendizaje al proporcionar retroalimentación (Trabelsi, Alnajjar, Ambali Parambil, Gochoo, & Ali, 2023).



Ilustración 7: Captura de pantalla de asistente de enseñanza. Tomado de (Trabelsi, Alnajjar, Ambali Parambil, Gochoo, & Ali, 2023)

5.1.4. ARQUITECTURAS ESCALABLES EN LA NUBE Y SERVERLESS

Un factor decisivo a la hora de pensar en una arquitectura de una solución digital es la escalabilidad. Esta dicta lo bien que responde un sistema cuando la carga aumenta y su estabilidad cuando se somete a dichas cargas. A medida que una solución digital crece, su complejidad aumenta en términos de rendimiento, seguridad y gestión. Una solución a estos problemas se basa en usar un diseño

modular, es decir, dividir el sistema en partes más pequeñas y sencillas. Esto hace que sea más fácil de gestionar y asegurar. Para mejorar la escalabilidad en estos diseños modulares se utiliza la técnica del escalado horizontal, el cual consiste en añadir sistemas de manera distribuida para manejar un aumento de carga. Esto implica que la aplicación deba construirse con este patrón en mente.

El uso de servicios de nube manejados, o “serverless”, permiten manejar la complejidad asociada a la gestión de múltiples componentes que están desplegados independientemente, a escala.

A fin de lograr una escalabilidad óptima, se debe tener en cuenta la seguridad y la tolerancia a fallos (Shawi, 2020).

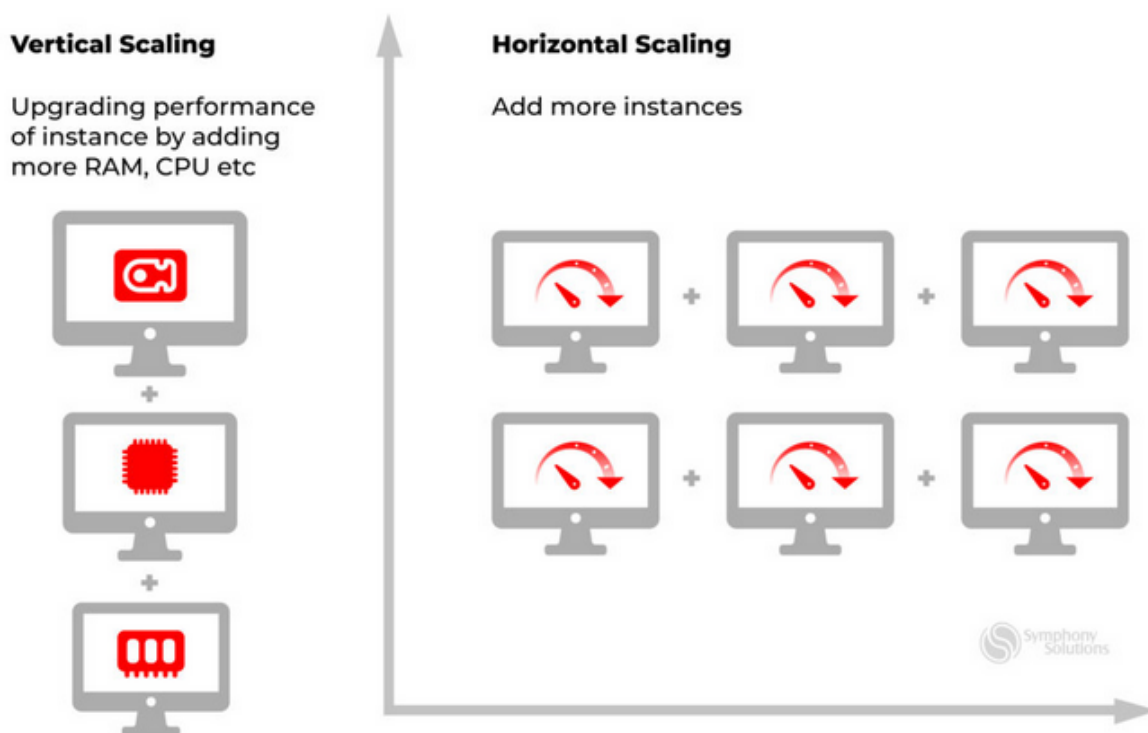


Ilustración 8: Tipos de escalabilidad en la nube. Tomado de (Symphony Solutions, 2020)

Las arquitecturas “serverless” proporcionan una manera de construir aplicaciones sin tener que administrar su infraestructura. Esto implica encargarse del aprovisionamiento de recursos, escalamiento y mantenimiento, y un modelo de costo de “pago por uso”. Los principales motivos por los que las empresas adoptan este modelo son por reducción de costos, escalabilidad y productividad de los desarrolladores.

Sin embargo, existen preocupaciones en torno a la seguridad, miedo a lo desconocido y falta de personal calificado. A pesar de sus ventajas, las arquitecturas

“serverless” todavía necesitan un equipo experimentado para que esta herramienta sea productiva y competente.

La principal ventaja que ofrecen estas arquitecturas es que los desarrolladores pueden concentrarse más en el desarrollo del producto sin preocuparse por administración u operación de servidores. También ofrecen opciones de observabilidad en la nube, despliegues más ágiles, flexibilidad y reducción de costos.

Entre sus limitaciones se encuentra que las aplicaciones con larga duración llegan a ser más costosas bajo este paradigma de arquitectura. También se hace un compromiso con el proveedor de nube, o “vendedor lock-in”. Además, se tiene el fenómeno de “arranque en frío” o “cold-start”, el cual consiste en un tiempo extra de ejecución para la primera invocación mientras que el proveedor aprovisiona los recursos necesarios.

Es importante entonces usar las arquitecturas serverless de manera adecuada para no incurrir en “anti-patrones” que generen sobrecostos, o que impacten negativamente la solución tecnológica (Fee, 2020).

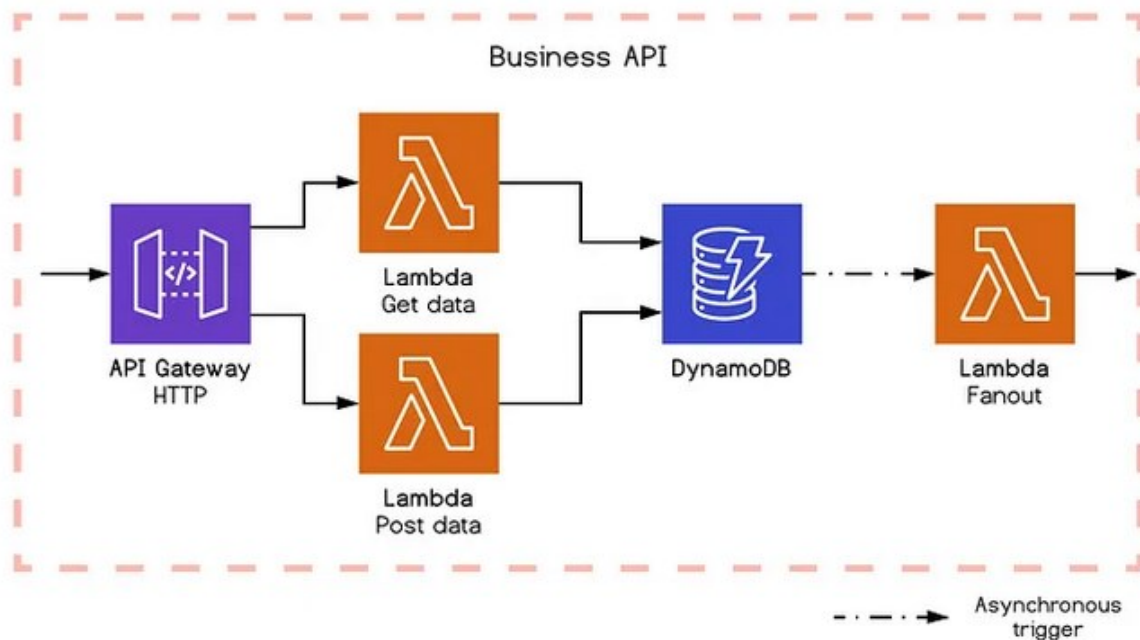


Ilustración 9: Ejemplo de una arquitectura Serverless para un servicio. Tomado de (Lefèvre, 2020)

Entre los modelos de desarrollo de software, encontramos la arquitectura monolítica y la arquitectura de microservicios. En la arquitectura monolítica, una sola base de código se encarga de manejar diferentes funcionalidades. Los componentes de software son dependientes entre sí y realizar un cambio en alguna funcionalidad puede ser restrictivo y requerir mucho tiempo debido a que se afecta la base de código de toda la solución.

Por otro lado, la arquitectura de microservicios busca dividir las funcionalidades en componentes independientes con alcance definido. Dichos servicios se comunican a través de interfaces y contratos establecidos, pudiendo ser actualizados, modificados y tener una flexibilidad en cuanto a escalabilidad.

Las aplicaciones monolíticas consisten en una interfaz de usuario, una base de datos y una aplicación en servidor. Todas estas capas se construyen bajo una misma base de código fuente. Para la arquitectura de microservicios, cada servicio es creado para que maneje una funcionalidad específica de negocio o lógica, y usa el patrón de “API” para comunicarse con otros actores.

La implementación de aplicaciones monolíticas es sencilla pero limitada debido a su restricción de tener una sola base de código fuente. Para los microservicios el enfoque de implementación es más complejo debido a que cada servicio se despliega por separado.

La depuración o rastreo de errores es otro aspecto que puede resultar más complejo en sistemas con arquitecturas monolíticas, debido a su naturaleza de separación. Esto implica, por lo general, hacer una búsqueda en varios servicios.

En términos de escalabilidad, las aplicaciones monolíticas obligan a escalar toda la solución tecnológica como una sola unidad, lo cual en muchos casos puede ocasionar que algunas funcionalidades estén sobredimensionadas. En los microservicios, el escalamiento de cada servicio depende de los requerimientos de ese servicio, por lo que se puede escalar tanto como sea necesario según las necesidades de cada servicio en particular.

En una etapa inicial, la arquitectura monolítica tiene una complejidad menor y puede resultar más atractiva para soluciones simples, sin embargo, a medida que aumenta el tamaño y la carga de la solución, los microservicios pueden ser más rentables debido a su capacidad de flexibilidad, mantenimiento y adaptabilidad. (Amazon Web Services, 2023).

En una arquitectura moderna de soluciones de software, la resiliencia y la escalabilidad son dos pilares fundamentales que permiten que las aplicaciones se ajusten a la demanda y resistan interrupciones de servicio.

La escalabilidad se define como la capacidad de un sistema de responder a diferentes cargas de trabajo, agregando o eliminando recursos del sistema.

Existen diferentes motivaciones y limitaciones para mejorar la escalabilidad de un sistema. Algunas de las motivaciones son optimizar costos y disminuir consumo de recursos, minimizar tiempo de inactividad de la aplicación, y asegurar que la demanda pueda suplirse durante los períodos de alto uso. En el lado del desarrollo del sistema, otro impulsor es minimizar el tiempo invertido en investigar fallos y administración de los sistemas, y aumentar el tiempo usado para generar nuevas características. Algunas de las limitaciones que pueden existir es hardware o

software difícil de operar y escalar, restricción de licencias o falta de personal experimentado.

Para abordar un buen diseño de arquitectura para sistemas de software se debe tener en cuenta todo el ciclo de vida de la aplicación, desde el diseño de infraestructura, comunicaciones, arquitectura de la aplicación, opciones de almacenamiento, implementación y cultura organizacional.

Se identifican entonces tres patrones principales: automatización, bajo acoplamiento entre componentes y el diseño basado en datos. La automatización mejora consistencia y velocidad, mientras que minimiza los errores involuntarios causados por procesos manuales. El bajo acoplamiento entre componentes se refiere a tratar dichos componentes de manera interdependientes, lo que habilita su flexibilidad y resiliencia. Por último, el diseño basado en datos se trata de recopilar y analizar métricas que permitan entender el comportamiento de tu aplicación y de los usuarios, de manera que puedan tomarse decisiones apoyados en estos datos.

La automatización del aprovisionamiento de la infraestructura es altamente recomendada. Para esto, una técnica común y efectiva es la “infraestructura como código” (IaC por sus siglas en inglés). Esto permite tratar la infraestructura de una manera similar que el código de la aplicación, permitiendo que el aprovisionamiento de recursos y la configuración se almacenen de manera que se pueda auditar y versionar. Esto minimiza errores humanos y mejora la consistencia y reproducibilidad de los entornos digitales.

Otra ventaja que brinda la adopción de la infraestructura como código es que habilita la inmutabilidad en la infraestructura. Esta requiere que los recursos que se crean no puedan ser modificados, si no que, al modificar la configuración de este en el repositorio, se validen los cambios y el recurso sea recreado a partir de la nueva configuración. La implementación de este patrón ayuda a tener procesos de despliegues más predecibles y controlados, lo cual es crucial para una aplicación escalable y resiliente.

La alta disponibilidad de una aplicación de software se mide por el tiempo en que un servicio es utilizable. Es importante diseñar las soluciones de software para que tengan una alta disponibilidad. Esto se logra haciendo una distribución de los recursos de cómputo, balanceo de carga, y la replicación de datos.

Los proveedores de nube públicas tienen diferentes opciones para distribuir físicamente los recursos utilizados en varias regiones alrededor del mundo. Al desplegar recursos en múltiples regiones, se logra la redundancia, la cual se define como la duplicación de componentes en un sistema. El tener redundancia en los recursos de una solución tecnológica permite que los servicios resistan de una mejor manera las interrupciones en el servicio de una zona o región en particular, mejorando la disponibilidad de la aplicación.

El balanceo de carga se refiere a la técnica que permite tener un grupo de recursos para distribuir el tráfico entre ellos, evitando que algunos componentes dentro de ese grupo se sobrecarguen mientras que otros permanecen ociosos.

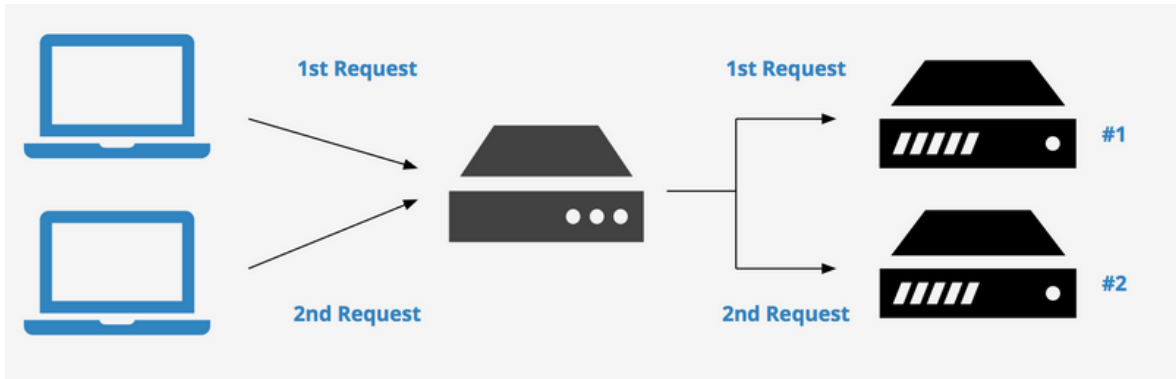


Ilustración 10: Ejemplo de balanceo de carga para una aplicación de servidor. Tomado de (Key CDN, 2023)

Los servicios manejados, o “serverless”, tienen incorporado redundancia, al mismo tiempo que reducen el tiempo que se dedica a gestionar la infraestructura. También, proveen balanceo de carga de manera nativa y automática, con poca o nada de configuración. Esto ayuda a alcanzar los objetivos de disponibilidad que se tengan.

Para poder mejorar la resiliencia, escalabilidad y disponibilidad de una aplicación, es importante entender el comportamiento. Tener a disposición un conjunto de datos que indiquen el comportamiento de una aplicación es importante para generar métricas, alarmas y registros de lo que ocurre en el sistema. Las métricas también pueden usarse para controlar o automatizar el escalamiento de los servicios. Entre los tipos de observación que pueden darse en una aplicación se encuentran:

- Monitoreo a nivel de aplicación.
- Monitoreo a nivel de servicio.
- Monitoreo de extremo a extremo.

Otro punto clave para diseñar una buena arquitectura para una aplicación de software es determinar una estrategia para el auto-escalado de los recursos. Esta estrategia dicta las reglas para agregar o reducir recursos computacionales para suplir la demanda. Hacer una buena estimación de la estrategia de auto-escalado para una aplicación es clave para equilibrar costos y experiencia de usuario, sin sobredimensionar recursos.

El auto-escalamiento es una funcionalidad que ajusta de manera automática los recursos de un sistema basado en métricas o condiciones. Los servicios manejados, o “serverless”, proveen automáticamente la capacidad de escalar rápidamente y reducir los servicios cuando ya no son necesarios.

Diseñar la arquitectura de una aplicación para que sea modular ayuda a aumentar la flexibilidad y la resiliencia, ya que se habilita el despliegue, gestión y escalamiento de manera independiente por cada componente. Tener servicios independientes y con bajo acoplamiento permite que diferentes equipos sean responsables por diferentes servicios y habilita el uso de diferentes tecnologías para construir dichos servicios.

Para diseñar una aplicación escalable, es crucial adoptar el patrón de “sin-estado” (en inglés, “stateless”). Este dice que no se retienen datos persistentes locales. Esto permite que los recursos se puedan añadir de manera que ayuda con la escalabilidad y recuperabilidad del servicio (Google Cloud Architecture Center, 2022).

Cuando se habla de serverless, nos referimos a servicios que permiten a los desarrolladores de software construir y ejecutar código sin tener que provisionar o administrar servidores o infraestructura. Esto libera a los desarrolladores de estas tareas y les permite enfocarse en escribir el mejor código de aplicación y plantear lógica de negocio que cumpla de la mejor manera posible los requisitos de una solución dada (IBM, 2023). Estos servicios serverless ofrecen un rendimiento alto y flexibilidad a escala, transfieren cargas de administración al proveedor de nube y genera equipos de trabajo más productivos (IBM Market Development & Insights, 2021).

En el espacio de computación en la nube se tienen varios conceptos que dictan la arquitectura a usar en determinada solución. En específico, se tienen los conceptos de “FaaS” (Functions as a Service) y “PaaS” (Platforms as a Service). La elección entre uno u otro tipo de servicio de nube debe basarse entre control y facilidad de uso. La arquitectura de la aplicación dicta también cual es más conveniente (De Rooms, 2020).

- **FaaS:** Se crean funciones las cuales funcionan como unidades de computo que ejecutan el código de la aplicación. Requiere que se descomponga la aplicación en microservicios aislados y desplegados de manera individual. Tienen escalabilidad automática inherente, generando nuevas funciones a medida que sean necesarios de acuerdo a la demanda. Esto evita cuellos de botella pero genera complejidad adicional ya que se construyen muchos componentes pequeños que deben orquestarse de manera adecuada para poder comunicarse y cooperar entre ellas.
- **PaaS:** Se refiere a plataformas que pueden alquilarse para desplegar una aplicación completa donde, a menudo, se vuelven más fáciles de configurar y preparar inicialmente. Las aplicaciones en PaaS suelen ser monolitos que escalan añadiendo otras instancias como “nodos”. Esto puede impedir un escalamiento en los momentos más críticos de demanda. Por lo general, los proveedores de PaaS terminan siendo más costosos a medida que la carga aumenta.

Para la comunicación entre microservicios, se encuentra que el patrón “Fan-Out” es útil cuando un servicio debe enviar solicitudes de mensajes a varios servicios. Es decir, una única solicitud debe poder procesarse por diferentes servicios. Las mejores prácticas indican que el mensaje debe originarse y propagarse a diferentes servicios, sin embargo, para tener una arquitectura desacoplada, dicho mensaje debe entregarse a los consumidores sin que el servicio que generó el mensaje deba mantener un listado de consumidores. Cada servicio suscriptor se suscribe a un tema o “topic”. Siguiendo las mejores prácticas, se establece que cada consumidor haga uso de una cola de mensajes, de manera que el mensaje que se transmita a través del tema sea entregado a diferentes colas de mensajes, y luego cada consumidor final recibe el mensaje de su respectiva cola. Esto permite tener resiliencia y desacoplamiento entre los productores y consumidores de mensajes (Richardson & Fröhner, 2019).

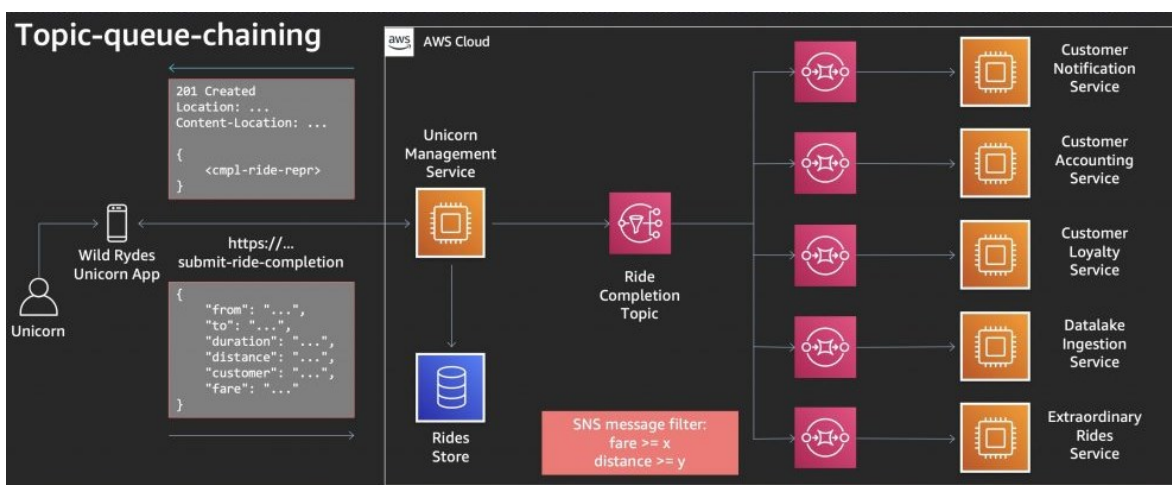


Ilustración 11: Arquitectura para una aplicación de ejemplo que usa el patrón "Fan-Out" para su comunicación asíncrona. Tomado de (Richardson & Fröhner, 2019)

5.1.5. MODELOS DE LENGUAJE GENERATIVOS

A mediados de 2021, OpenAI, una compañía especializada en la investigación y desarrollo de inteligencia artificial (Open AI, 2023), lanzó al público su último avance técnico en un modelo de lenguaje capaz de generar texto en lenguaje natural casi indistinguible de texto generado por humanos (Dale, 2021).

Los sistemas inteligentes hoy en día ofrecen capacidades que se basan en el aprendizaje de máquina (Machine Learning). Este es un método por el cual un sistema aprende patrones a partir de un entrenamiento previo. El aprendizaje profundo (Deep Learning) es un subconjunto dentro del aprendizaje de máquina que

usa técnicas más avanzadas basado en redes de neuronas artificiales para lograr una mejor inferencia y aprendizaje. Estas técnicas permiten construir sistemas inteligentes (no-determinísticos) que, en algunos casos, superan el desempeño humano (Janiesch, Zschech, & Henrich, 2021).

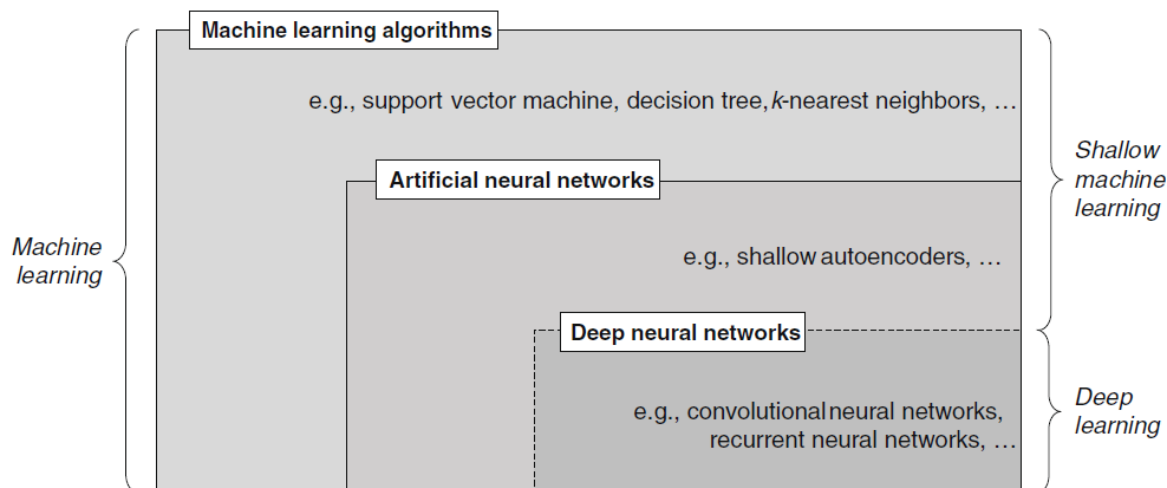


Ilustración 12: Diagrama de Venn sobre conceptos de aprendizaje de máquina (Machine learning) y las diferentes clases que se presentan. Tomado de (Janiesch, Zschech, & Henrich, 2021)

En el campo de procesamiento de lenguaje natural, uno de los objetivos es que un sistema computacional logre comprender e interactuar diferentes tipos de entrada como texto o voz, de manera natural, como lo haría un humano. Dada la alta complejidad de esta tarea, se han encontrado avances en esta área al implementar técnicas de aprendizaje profundo (Deep Learning) en algunas tareas como etiquetado de parte del discurso, segmentación y reconocimiento de patrones y roles. Aplicar estos métodos hace que el sistema aprenda representaciones internas, a menudo de datos sin etiquetar (Collobert, et al., 2011).

OpenAI fue una compañía fundada a finales de 2015 como una organización sin ánimo de lucro. Su principal objetivo es investigar y desarrollar la inteligencia artificial general (AGI), y sus beneficios para la humanidad. En febrero de 2019 anunciaron “GPT-2”, un modelo generativo entrenado en 40GB de texto. La manera como funciona este tipo de modelos es la de predecir la siguiente palabra en un texto basado en su contexto anterior, de manera que el resultado tenga sentido y sea coherente (Dale, 2021).

En un informe técnico de OpenAI sobre cómo mejorar el entendimiento del lenguaje por medio de preentrenamientos generados (Radford, Narasimhan, Salimans, & Ilya, 2019), se detalla el proceso que siguen los investigadores y desarrolladores para mejorar el rendimiento de sus modelos generativos, como GPT.

Los métodos de aprendizaje de profundo (Deep Learning) supervisado aplicados al procesamiento de lenguaje natural requieren grandes cantidades de datos, los cuales deben ser etiquetados manualmente. Esto limita la aplicación de muchos

casos de uso basados en estos modelos supervisados. Por otro lado, los modelos con preentrenamiento no supervisado y ajuste fino supervisado, permiten al modelo aprender una gran cantidad de tareas con mínima adaptación (Radford, Narasimhan, Salimans, & Ilya, 2019).

En marzo de 2023, OpenAI anunció su más reciente y avanzado modelo, “GPT-4”. Así como sus predecesores GPT-3.5, GPT-3 y GPT-2, es un modelo basado en transformadores que le permite generar y entender texto de manera natural. GPT-4 es mucho más capaz que sus anteriores versiones, permitiéndole sobrepasar a muchos humanos en varias tareas. En pruebas diseñadas para humanos, se encontró que GPT-4 sobrepasaba la gran mayoría de humanos que las tomaban. Se caracterizó el modelo comparándolo con sistemas del estado del arte, encontrando que GPT-4 sobrepasaba dichos sistemas en rendimiento y precisión. (OpenAI, 2023).

	GPT-4 Evaluated few-shot	GPT-3.5 Evaluated few-shot	LM SOTA Best external LM evaluated few-shot	SOTA Best external model (incl. benchmark-specific tuning)
MMLU [49] Multiple-choice questions in 57 subjects (professional & academic)	86.4% 5-shot	70.0% 5-shot	70.7% 5-shot U-PaLM [50]	75.2% 5-shot Flan-PaLM [51]
HellaSwag [52] Commonsense reasoning around everyday events	95.3% 10-shot	85.5% 10-shot	84.2% LLaMA (validation set) [28]	85.6 ALUM [53]
AI2 Reasoning Challenge (ARC) [54] Grade-school multiple choice science questions. Challenge-set.	96.3% 25-shot	85.2% 25-shot	85.2% 8-shot PaLM [55]	86.5% ST-MOE [18]
WinoGrande [56] Commonsense reasoning around pronoun resolution	87.5% 5-shot	81.6% 5-shot	85.1% 5-shot PaLM [3]	85.1% 5-shot PaLM [3]
HumanEval [43] Python coding tasks	67.0% 0-shot	48.1% 0-shot	26.2% 0-shot PaLM [3]	65.8% CodeT + GPT-3.5 [57]
DROP [58] (F1 score) Reading comprehension & arithmetic.	80.9 3-shot	64.1 3-shot	70.8 1-shot PaLM [3]	88.4 QDGAT [59]
GSM-8K [60] Grade-school mathematics questions	92.0%* 5-shot chain-of-thought	57.1% 5-shot	58.8% 8-shot Minerva [61]	87.3% Chinchilla + SFT+ORM-RL, ORM reranking [62]

Ilustración 13: Rendimiento de GPT-4 en pruebas académicas comparado con otros modelos similares del estado del arte. Tomado de (OpenAI, 2023)

Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)

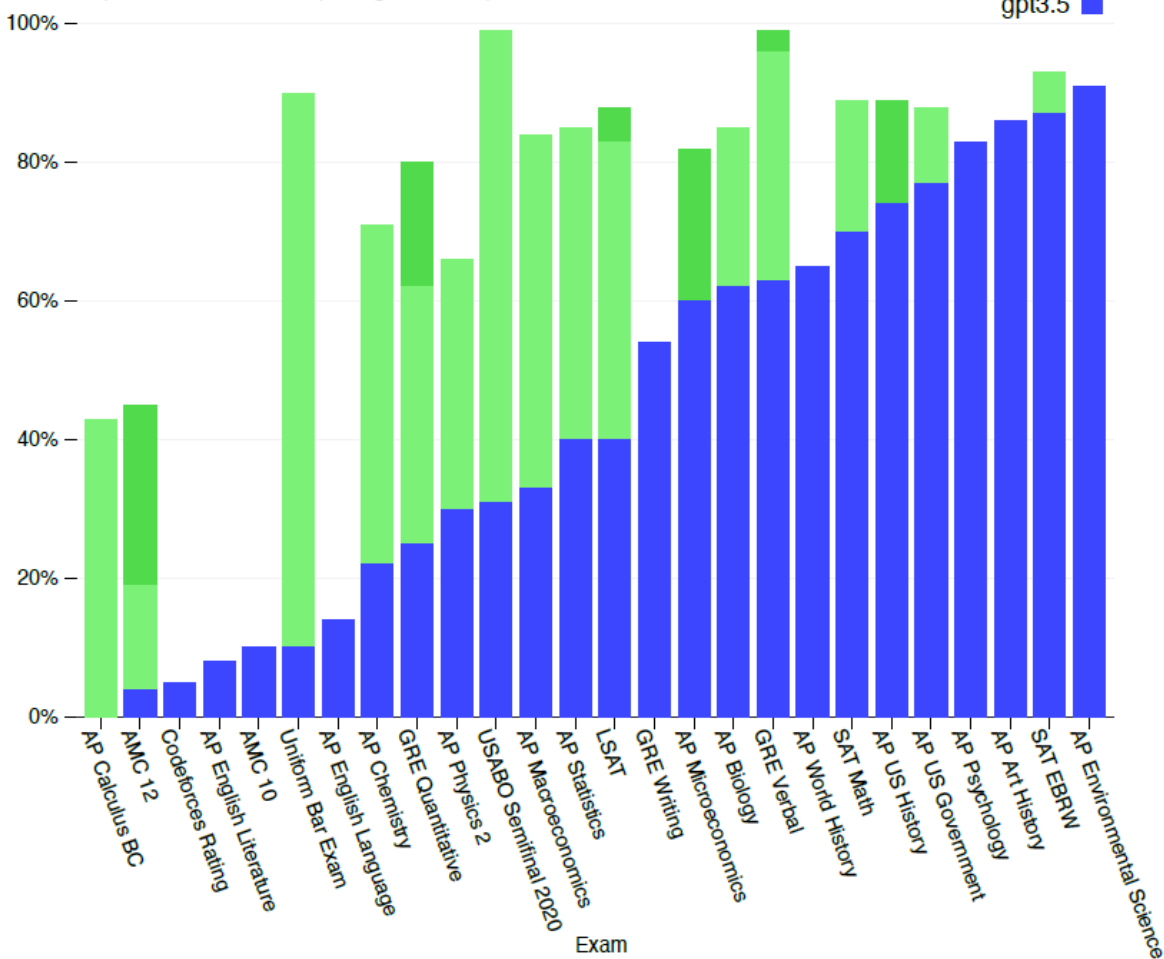


Ilustración 14: Rendimiento de modelos GPT en diferentes pruebas académicas. Tomado de (OpenAI, 2023)

Aunque el desempeño de estos modelos generativos sobrepasa inclusive el desempeño de un humano, traen consigo varios factores para tener en cuenta. Algunos de esos factores son:

- **Alucinaciones:** Se refiere a cuando el texto generado por el modelo no tiene sentido o no es cierto en relación con su fuente o contexto (OpenAI, 2023). GPT-4 fue entrenado para reducir la tendencia a generar contenido sin sentido a partir de datos obtenidos de otros modelos
- **Contenido no-deseado:** Hace referencia a los intentos de usar el modelo para generar contenido que puede ser dañino, desinformativo, o con sesgos. GPT-4 reconoce y evita generar texto con relación a estos tipos de contenidos a nivel general (OpenAI, 2023).

OpenAI ha implementado medidas de seguridad en su más avanzado modelo “GPT-4” para reducir el contenido no deseado o sus alucinaciones, sin embargo, aún se puede ser vulnerable a ataques. El modelo está en constante monitoreo y evolución para continuar mejorando la seguridad y confianza que permitan una adopción segura y responsable de estos modelos (OpenAI, 2023).

5.2. ESTADO DEL ARTE

En esta sección se presenta una comparación de los trabajos revisados relacionados con la problemática que se quiere abordar en este trabajo. Este análisis recopila la información de otras investigaciones de manera visual, y como se relacionan con el trabajo realizado en este documento.

- **Referencia:** Indica la referencia del trabajo revisado.
- **Funcionalidad:** Dice la funcionalidad lograda en el trabajo.
- **Tipo:** Indica si el trabajo es Asíncrono o Síncrono:
 - **Síncrono:** Se refiere a que el experimento usa datos de sujetos de estudio en tiempo real. La observación y recopilación de datos se da de manera directa mientras los sujetos están realizando la actividad.
 - **Asíncrono:** Los datos para realizar el trabajo es información recopilada anteriormente.
- **Enfoque:** Hace referencia al enfoque u objetivo que se identifica del trabajo.
- **Observaciones:** Pequeña descripción del estudio y algunas percepciones relevantes para el trabajo de este documento.

Referencia	Funcionalidad	Tipo	Enfoque	Observaciones
(Barton Essel, Vlachopoulos, Tachie-Menson, Eduafua Johnson, & Kwame Baah, 2022)	Chatbot para apoyar proceso de enseñanza	Asíncrono	Apoyar enseñanza de alumnos de manera asíncrona	Estudio enfocado en chatbots asíncronos para reforzar la educación de estudiantes a distancia. El personal académico debe personalizar e integrar el chatbot para sacar mayor provecho de este
(Madsen, Júlio, Gucik, Steinberg, & Parra, 2020)	Medición de nivel de atención a través de movimientos oculares	Síncrono	Relacionar movimientos oculares con nivel de atención y resultados académicos	El estudio se explora como se relacionan los movimientos oculares de alumnos que ven un video con su nivel de atención y sus resultados académicos en pruebas.
(Trabelsi, Alnajjar, Ambali Parambil, Gochoo, & Ali, 2023)	Modelo que identifica acciones en estudiantes	Síncrono	Proveer al docente una ayuda sobre el comportamiento	Se creó un modelo de aprendizaje de máquina basado en visión que permite monitorear la participación, nivel de atención y emociones de los estudiantes

			de sus estudiantes	en una clase presencial. El modelo pretende ayudar a los instructores en su pedagogía y los resultados muestran una precisión del 76%
(Wang, 2021)	Modelo basado en análisis de características faciales para medir y atención y aprendizaje	Asíncrono	Identificar comportamientos de aprendizaje en línea a través de reconocimiento de imágenes	Esta investigación busca superar barreras en la educación en línea en términos de la identificación de comportamientos de aprendizaje y evaluar sus efectos en la enseñanza. Se extraen imágenes de los rostros de los alumnos de un video de una clase en línea y realiza el análisis basado en estos cuadros
(Tabassum, Allen, & De, 2020)	Modelo de clasificación de estudiantes atentos y relación con emociones	Asíncrono	Identificar atención y desatención en estudiantes mientras toman sus clases, y relacionar con sus emociones faciales	El trabajo consiste en varias etapas, en la primera etapa capturan datos de los estudiantes mientras reciben sus clases. Se etiquetan los estudiantes como atentos o desatentos y, a partir de allí, se entrena un modelo usando imágenes de los estudiantes. Luego, realizan una correlación del nivel de atención de un estudiante y sus emociones capturadas por "AWS Rekognition"
(Cheng, Chien, & Huang, 2017)	Modelo de identificación de atención en tiempo real usando detección de ondas cerebrales y cámara web	Síncrono	Mejorar el reconocimiento por parte de un profesor del nivel de atención de sus estudiantes	El estudio se basa en usar identificación de nivel de atención basado en datos proveídos por seguimiento de movimientos oculares usando una cámara web, y ondas cerebrales captadas por un dispositivo en el estudiante. Luego, el sistema entrega una retroalimentación al estudiante para corregir el comportamiento en caso de que sea necesario.
(Terraza, et al., 2022)	Modelo de reconocimiento de atención basado en posición dirección del rostro	Síncrono	Ayudar al educador a tener una mejor visión de los niveles de atención de sus estudiantes	En este trabajo se identifica un estudiante atento o desatento basado en puntos clave de su rostro. Si dicho rostro se encuentra en determinadas direcciones se identifica como desatento. Luego se entrenó un modelo de aprendizaje de máquina que mostró precisiones de entre 90.65 a 95.14%.

Tabla 1: Comparación de trabajos revisados como parte del estado del arte. Creación propia

6. DESARROLLO DEL TRABAJO

En esta sección se plantea el diseño e implementación de un prototipo de asistente virtual que sea capaz de capturar y visualizar el nivel de atención de los estudiantes de forma no intrusiva. Específicamente, se aborda el desarrollo de dicho prototipo definiendo los requisitos funcionales y no funcionales, la arquitectura de la solución, los casos de uso, y la arquitectura de los sistemas necesarios. Se muestra también elecciones de tecnologías y herramientas para la construcción del prototipo, se da un diseño detallado usando el “modelado C4”, se muestra cómo se implementa la captura del nivel de atención y como se generan las recomendaciones de actividades de aprendizaje activo al profesor, teniendo en cuenta sus vulnerabilidades y protecciones necesarias.

De acuerdo con la necesidad de investigar la utilidad de una herramienta tecnológica que permita monitorear el nivel de atención de estudiantes en una clase virtual, se propone desarrollar el prototipo de un sistema de asistente virtual que sea capaz de capturar y mostrar el nivel de atención de los estudiantes en una clase virtual de manera no-intrusiva para la clase y con mínima intervención por parte del profesor o estudiantes.

El prototipo debe ser extensible para que pueda ser adoptado e implementado por diferentes instituciones educativas, sin que esto comprometa su rendimiento o funcionamiento.

A continuación, se presentan las consideraciones para que el prototipo cumpla su objetivo y permita resolver la pregunta de investigación:

- **CON-1:** La herramienta debe tener la capacidad de medir y monitorear el nivel de atención de los estudiantes en una clase virtual sin que esto implique interrumpir la sesión.
- **CON-2:** La herramienta debe tener la capacidad de mostrar los resultados que obtiene de manera visual a medida que la clase virtual transcurre.
- **CON-3:** La herramienta debe tener la capacidad de almacenar las mediciones por alumno y por clase.
- **CON-4:** La herramienta debe ser diseñada para que pueda ser usada en diferentes clases e instituciones educativas, de manera simultánea, sin que esto afecte de manera negativa su funcionamiento.

Teniendo en cuenta estas consideraciones iniciales, se identifica que el prototipo debe ser una herramienta no intrusiva que permita medir el nivel de atención de los estudiantes en una clase virtual. Para lograr este cometido, se limita el uso del prototipo a clases virtuales realizados por medio de videollamadas en las plataformas de llamadas y conferencias virtuales más comunes hoy en día.

6.1. REQUISITOS

Un producto de software requiere una planificación cuidadosa para poder tener éxito y evitar reprocesos. Dentro de esta planificación, una de las primeras etapas es la especificación de requisitos que debe cumplir el producto. Sin embargo, la comunicación efectiva entre los desarrolladores de software y los interesados es un reto. Es necesario tener un equilibrio entre la perspectiva de negocio y lo técnico para evitar fallos en los proyectos de software. Es preferible distribuir la toma de decisiones a lo largo del desarrollo de un proyecto, en lugar de hacerlo desde el principio del proyecto. Las historias de usuario son una herramienta valiosa para lograr este propósito. Estas historias de usuario proporcionan una descripción de las funcionalidades del producto requeridas y facilitan conversaciones para detallarlas. Se deben representar funcionalidades valoradas y evitar detalles técnicos irrelevantes para las mismas. Lo importante es escribir historias que aporten valor. (Cohn, 2004).

6.1.1. Requisitos funcionales

Los requisitos funcionales se definen a manera de historia de la siguiente forma:

El profesor entra en una clase virtual junto con sus estudiantes a través de una plataforma digital. Dado que no se está en la misma habitación, se le dificulta al profesor seguir el nivel de atención de sus estudiantes.

Dada la importancia de que el profesor conozca el nivel de atención de los estudiantes en una clase virtual, se crea la necesidad de conocer esta información por medios tecnológicos, de manera que no interrumpan la clase ni distraigan a los alumnos.

Para dar respuesta a esta necesidad, se propone la creación de una herramienta tecnológica para el profesor que logre suplir la necesidad expuesta.

La herramienta tecnológica debe ser capaz de capturar y presentar el nivel de atención de los estudiantes en una clase virtual. La información debe presentarse de manera que el profesor pueda usarla y entenderla, para poder adaptar el proceso pedagógico acorde. Además, la herramienta tecnológica no debe interrumpir o distraer a los alumnos.

La herramienta tecnológica debe ser capaz de almacenar la información que captura para que se pueda analizar en el futuro y sea útil para profesores, alumnos e instituciones educativas.

La herramienta tecnológica debe ser capaz de proveer sugerencias de actividades de aprendizaje activo al profesor que sean relevantes para la clase, basado en el nivel de atención de los estudiantes. Dichas sugerencias deben ser tomando las últimas mediciones únicamente y no todas las mediciones a lo largo de la clase.

De lo anterior, los actores identificados que interactúan con la herramienta tecnológica son los siguientes:

- **Profesores**
- **Alumnos**
- **Instituciones educativas**

Con base en lo anterior y con el fin de ajustar los detalles que debe cumplir el prototipo, se definen las siguientes historias de usuario:

- **HU-1:** Como Profesor, quiero recibir información sobre el nivel de atención de los estudiantes, para poder adaptar mis métodos de enseñanza basado en esa información.
- **HU-2:** Como Profesor, quiero que la información sobre el nivel de atención de los estudiantes sea capturada de manera no intrusiva, para que no interrumpa el flujo normal de la clase.
- **HU-3:** Como Profesor, quiero que la información sobre el nivel de atención de los estudiantes sea presentada de manera visual y legible, para que pueda interpretarla de manera fácil.
- **HU-4:** Como Profesor, quiero que se capture la información sobre el nivel de atención de los estudiantes de manera individual por cada estudiante y de manera ponderada de toda la clase, para poder adaptar mis métodos de enseñanza basado en esa información.
- **HU-5:** Como Profesor, quiero que la información que se presente sobre el nivel de atención de los estudiantes sea actual, para que pueda tomar adaptar el proceso de enseñanza basado en información actual.
- **HU-6:** Como Profesor, quiero que se almacene la información sobre nivel de atención de mis estudiantes por cada clase, para que pueda ser consultada y analizada.
- **HU-7:** Como Profesor, quiero que se hagan recomendaciones sobre actividades de aprendizaje activo basado en el nivel de atención de mis estudiantes, para que pueda adaptar mis métodos de enseñanza.
- **HU-8:** Como estudiante, quiero que la información que se capture sobre el nivel de atención sea no intrusiva, para que no interrumpa mi proceso de aprendizaje.
- **HU-9:** Como Institución Educativa, quiero que la información que se capture sobre el nivel de atención de los estudiantes se pueda consultar, para poder evaluar y analizar el proceso de aprendizaje y enseñanza de estudiantes y profesores.

6.1.2. Requisitos no funcionales

Es importante definir los requisitos no funcionales para un determinado sistema de software. Estos abordan temas como usabilidad, flexibilidad, rendimiento, seguridad, entre otros. Aunque los requisitos funcionales han recibido mayor atención, los requisitos no funcionales permiten tener confianza sobre la utilidad de la solución de software. Estos requisitos no funcionales deben tenerse en cuenta desde el principio, ya que permiten entender cómo debe comportarse el sistema y cómo desempeñar la función para la que fue creado. Además, los requisitos no funcionales aportan una perspectiva que ayuda a los ingenieros de software a entender completamente la solución a construir. (Chung & Sampaio do Prado Leite, 2009).

A continuación, se procede a generar los requisitos no funcionales que debe cumplir la herramienta tecnológica:

- **Disponibilidad:** La herramienta tecnológica debe estar responsivo y en funcionamiento durante las clases virtuales que lo usan.
- **Privacidad y Protección de datos:** La herramienta tecnológica recopila y almacena datos, los cuales deben estar protegidos en tránsito y en reposo. Asimismo, los datos recolectados no deben ser utilizados para ningún otro propósito.
- **Escalabilidad:** La herramienta tecnológica debe ser capaz de soportar un número creciente de alumnos, profesores y clases virtuales sin que esto comprometa su rendimiento.
- **Eficiencia:** La herramienta virtual debe ser capaz de capturar, procesar y analizar datos con la menor demora posible para que la retroalimentación proveída sea relevante en el momento.
- **Rendimiento:** La herramienta tecnológica debe ser capaz de manejar múltiples clases, profesores y alumnos de manera simultánea, sin que esto afecte la calidad de esta.
- **Mantenibilidad:** El código con el que está construida la herramienta tecnológica debe ser fácil de modificar y extender, para que sea sencillo corregir fallos o añadir nuevas características en el futuro.

6.2. ARQUITECTURA DE LA SOLUCIÓN

Dados los requisitos funcionales y no-funcionales descritos, se entiende que la construcción de la herramienta digital será usada, principalmente, por el profesor. Sin embargo, debe poder expandirse para ser usado simultáneamente por varios profesores en diferentes clases. Esto implica que el sistema debe adoptar

tecnologías escalables y elásticas, las cuales cumplen los objetivos de escalabilidad mientras que mantienen los costos bajos.

Con el fin de cumplir el objetivo de la solución de software descrita, se deben definir ciertas suposiciones que ayudarán a acotar la solución y dictarán el diseño y la arquitectura del sistema:

- **SUP-1:** La clase que el profesor dicte a los alumnos será virtual y sincrónica. Es decir, la clase no requiere tener cercanía o presencialidad, y dicha clase será dictada “en vivo”, no como una grabación.
- **SUP-2:** El profesor y los alumnos usarán la plataforma digital “Microsoft Teams” para conectarse de manera virtual por medio de una videollamada.
- **SUP-4:** Los profesores estarán conectados a través de sus propios medios tecnológicos a la clase virtual.
- **SUP-5:** Los medios tecnológicos que usen los alumnos y el profesor les permite escuchar a los demás participantes de la videollamada sin inconvenientes. Los alumnos podrán ver el contenido de la clase y el profesor podrá proyectar el contenido preparado sin problemas técnicos.
- **SUP-6:** El medio tecnológico que use cada alumno para recibir la clase virtual tiene la capacidad de capturar el video del rostro del alumno, y este video podrá transmitirlo a través de la videollamada sin inconvenientes técnicos.
- **SUP-7:** El profesor tendrá acceso a dos pantallas: Una pantalla para proyectar el contenido de la clase o leer las notas que haya preparado sobre el tema, y la otra pantalla para ver y analizar la retroalimentación sobre el nivel de atención.
- **SUP-8:** Los alumnos tendrán activado la cámara que captura video de su medio tecnológico encendida y su rostro estará en rango de alcance de dicha cámara.

En el estudio publicado por (Tabassum, Allen, & De, 2020), se realizó una captura del nivel de atención en estudiantes entrenando un modelo de aprendizaje de máquina. Luego, realizaron un análisis estadístico para correlacionar dicho nivel de atención con las emociones identificadas en los rostros de los estudiantes, dando como resultado una manera de calcular el nivel de atención de los estudiantes a partir de las emociones identificadas en sus rostros.

Dado que esta es una manera no-intrusiva de medir el nivel de atención de los estudiantes, se optó por adoptar esta técnica para medir dicho nivel de atención.

Teniendo en cuenta que las clases virtuales serán dictadas de manera síncrona y a través de la plataforma “Microsoft Teams”, es necesario añadir un participante adicional a la clase virtual que se encargue de tomar las imágenes de los rostros de todos los estudiantes, usando la característica de la plataforma que permite mirar la transmisión de video de los participantes en una sola pantalla. Dicha característica de la plataforma se conoce como “Vista de Galería” (Microsoft, 2023), la cual permite ver los videos que transmiten los participantes de la llamada. Si hay más de 10 participantes activos en la llamada, se puede activar la vista “Galería Grande”

(Microsoft, 2023). Con ayuda de su dispositivo, puede tomar capturas de pantalla en intervalos determinados para obtener todas las imágenes de los rostros de los participantes. Con un análisis posterior, se pueden identificar y extraer los rostros de manera individual, así como asignarles sus niveles de emoción.

En cuanto a la arquitectura de la herramienta tecnológica, la solución de software debe poder recibir tráfico y demanda por parte de usuarios. Los componentes y arquitectura del sistema se ven directamente afectados por donde estarán alojados los componentes de cómputo, almacenamiento, registro, entre otros. Para esto, dos opciones viables son alojamiento en servidores privados, o alojamiento en nube pública. A continuación, se exponen puntos a favor de las tecnologías de nube (Hazdun, 2023):

- Muchas empresas, tanto grandes como pequeñas, han encontrado beneficios en alojar sus datos en la nube. Les resulta más seguro y rentable que mantener un centro de datos privado.
- En cuanto a costos, los proveedores de servicios de nube cubren los costos de mantenimiento, reparación y actualización de hardware.
- La escalabilidad y elasticidad de los recursos de nube permite a las empresas optimizar costos ya que pueden planear bajo la demanda o, inclusive, usar servicios manejados o “serverless” para que esta tarea se haga automáticamente.
- Cuando se está iniciando la nube pública resulta más atractiva debido a que sus costos iniciales son bajos comparados con hacer una configuración local.

Sin embargo, también se identifican algunas desventajas que deben tenerse en cuenta y valorarse a la hora de escoger una arquitectura de solución basada en tecnologías en la nube:

- Riesgos de seguridad de los datos y dependencia de internet: Dado que los datos deben viajar a través de internet puede significar una desventaja en términos de fiabilidad y seguridad comparado con un modelo 100% local.
- Falta de control total sobre todos los recursos: En un contexto de computación en la nube, el equipo de TI de una empresa no tiene un control completo sobre la infraestructura de nube, ya que esta es responsabilidad del proveedor de nube.

En últimas la decisión dependerá del caso de uso y las funcionalidades esperadas de la solución. (Fisher, 2018).

En este caso, dadas las ventajas que traen los servicios de nube públicas como el bajo costo inicial, posibilidad de escalamiento horizontal y vertical, elasticidad, seguridad, entre otras; Hacen que sea una opción más atractiva para un prototipo que el alojamiento local tradicional. Con base en esto, se procede a diseñar una arquitectura de una solución alojada en la nube.

6.2.1. Casos de uso

Se identifican entonces los siguientes casos de uso o épicas que la solución debe satisfacer:

- Profesor:
 - Participar de la clase virtual.
 - Visualizar el nivel de atención de sus estudiantes mientras dicta la clase virtual.
 - Visualizar registros históricos del nivel de atención de las clases.
 - Recibir recomendaciones de actividades de aprendizaje activo mientras dicta la clase virtual.
- Alumnos:
 - Participar de la clase virtual.
- Participante de captura de pantalla:
 - Realizar capturas de pantalla del video de los rostros de los alumnos mientras estos recién las clases.

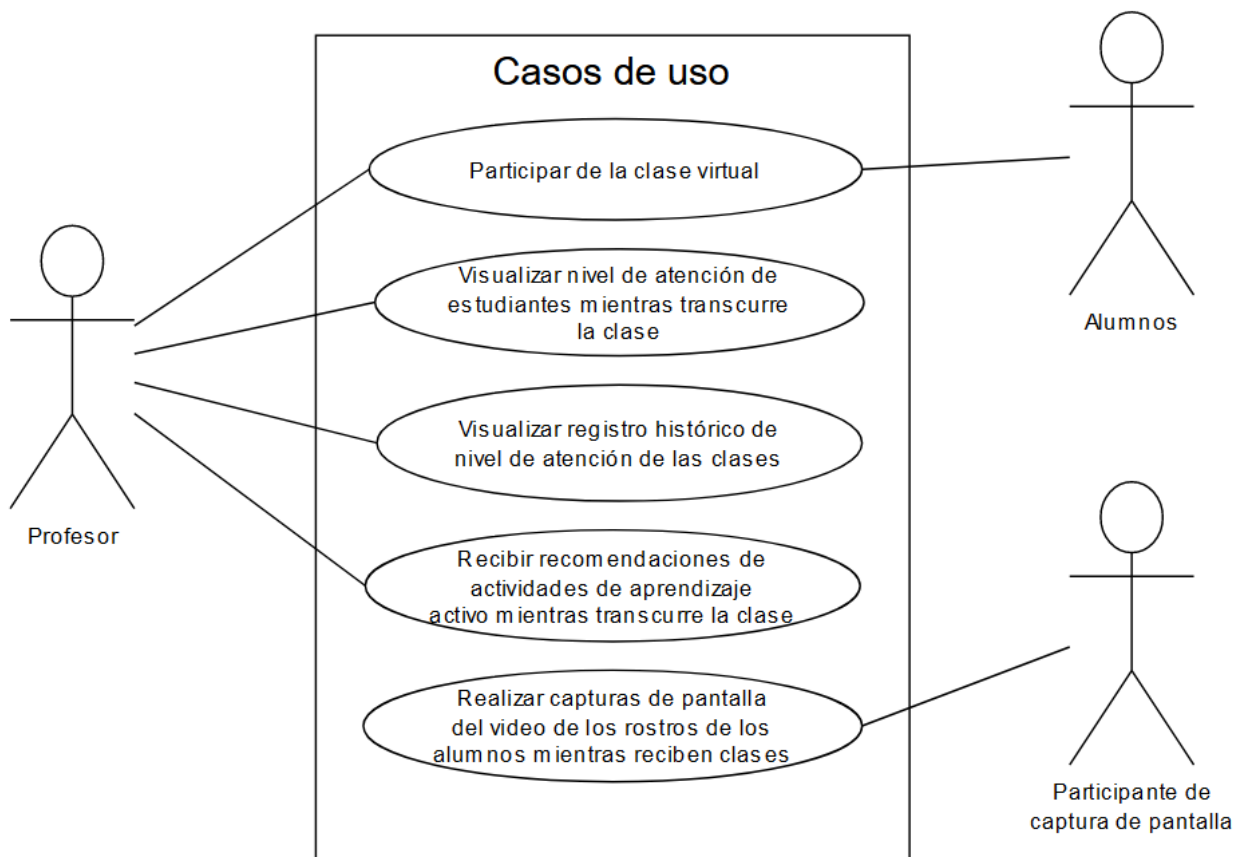


Ilustración 15: Diagrama de casos de uso de la solución. Creación propia

A partir de los requisitos, consideraciones y casos de usos expuestos hasta ahora, se define la necesidad de crear 3 sistemas dentro de la arquitectura de la solución, llamadas por sus nombres en inglés:

- **Screenshot Analyzer:** “Analizador de capturas de pantalla”, se encarga de hacer todo lo relacionado con capturar toma de pantalla de rostros de estudiantes, identificar rostros y calcular nivel de atención.
- **Teacher Assistant:** “Asistente del profesor”, es el sistema encargado de mostrar al profesor el estado actual del nivel de atención de la clase a medida que transcurre. También sirve para generar nuevas recomendaciones de actividades de aprendizaje activo para el profesor
- **Learning analytics:** Es el sistema que tiene como objetivo guardar datos históricos sobre el nivel de atención por alumno y por clase, Así como guardar cuando se generan recomendaciones de actividades de aprendizaje activo por parte del profesor. El profesor puede consultar esta información en el futuro para evaluar y analizar su pedagogía de enseñanza, y el desempeño de la clase y de sus alumnos.

Para entender mejor el rol de cada sistema y como son las interacciones entre los diferentes actores, sistemas y subsistemas, es útil presentar diagramas de secuencia para cada sistema (Larman, 2004).

6.2.2. Analizador de capturas de pantalla

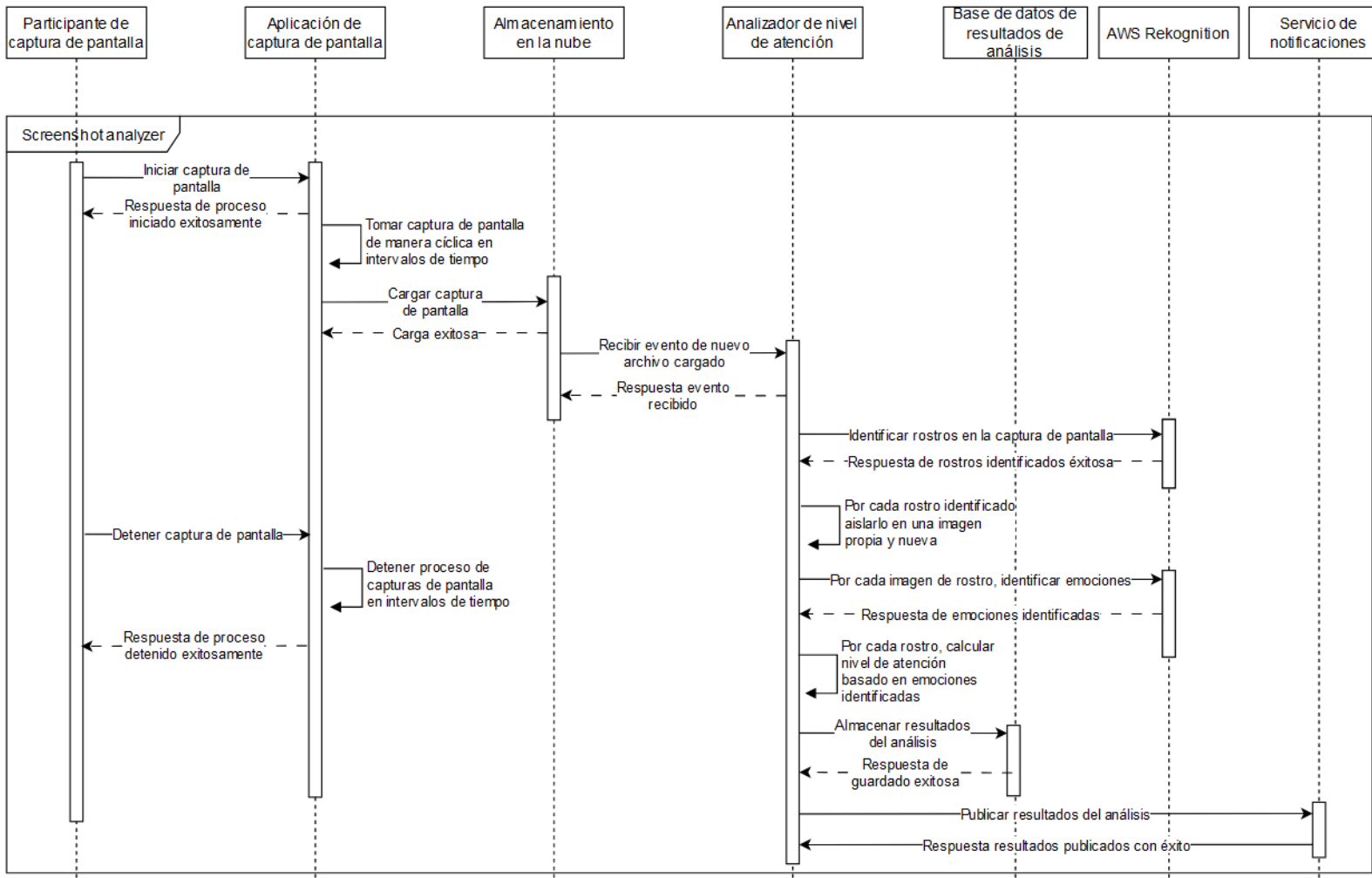


Ilustración 16: Diagrama de secuencia del sistema "Analizador de capturas de pantalla" o "Screenshot analyzer". Creación propia

El objetivo del analizador de capturas de pantalla "Screenshot Analyzer" es capturar los rostros de los estudiantes, analizarlos y calcular el nivel de atención de los rostros y de la clase. Se diseña de manera escalable y usando tecnologías de nube para almacenar las capturas de pantalla. También se diferencia el proceso de captura de pantalla con el análisis de la imagen, permitiendo un desacoplamiento entre ambos subprocesos y haciendo que el proceso de analizar las capturas de pantalla se vuelva asíncrono. Al finalizar el análisis, dado que se trata de un proceso asíncrono que puede ser usado por varios sistemas, se propone usar el patrón de "pub/sub" (Reselman, 2021) ("publisher/subscriber", o publicador y suscriptores por su nombre en inglés) para tener una solución de mensajería escalable y desacoplada.

- **Aplicación de captura de pantalla:** Esta es una aplicación que debe ser iniciada por el “participante de captura de pantalla” mientras esté dentro de la clase virtual. La aplicación toma capturas de pantalla a medida que la clase virtual transcurre y las carga a un almacenamiento en la nube.
- **Participante de captura de pantalla:** Hace referencia al participante de la clase virtual que no es un estudiante ni el profesor, cuyo propósito es ingresar a la misma clase virtual que ellos e iniciar el proceso de captura de pantalla, asegurándose de que su videollamada esté en modo “Galería”.
- **Almacenamiento en la nube:** Ese refiere a un servicio de almacenamiento de archivos en la nube que permita guardar las capturas de pantallas y permitan iniciar un proceso asíncrono de procesamiento de imágenes.
- **Analizador de nivel de atención:** Es una unidad de cómputo que recibe la referencia de una nueva imagen cargada y con ella puede realizar diferentes tareas, solicitar identificación y análisis de emociones, recortar rostros individualmente, calcular nivel de atención y publicar resultados.
- **AWS Rekognition:** Es un sistema externo de nube que usa técnicas de aprendizaje de máquina para realizar análisis de imágenes y videos. Provee un rango considerable de servicios relacionados con análisis de imágenes y videos. Para el caso de uso de la solución tecnológica en cuestión, las operaciones a usar son identificar y guardar rostros, y analizar emociones en rostros de personas (Amazon Web Services, 2023).
- **Base de datos de resultados de análisis:** Es la base de datos usada para almacenar los resultados de cada análisis que se finalice. Es el punto de partida de otros sistemas que quieran consultar más detalles de un análisis en particular.
- **Servicio de notificaciones:** Hace referencia al servicio de salida del proceso asíncrono. Una vez finalizado el análisis, se notifica a los demás sistemas que interesados sobre el nuevo análisis generado. Dichos sistemas interesados reciben un mensaje el cual cada sistema maneja de manera individual.

6.2.3. Asistente del profesor

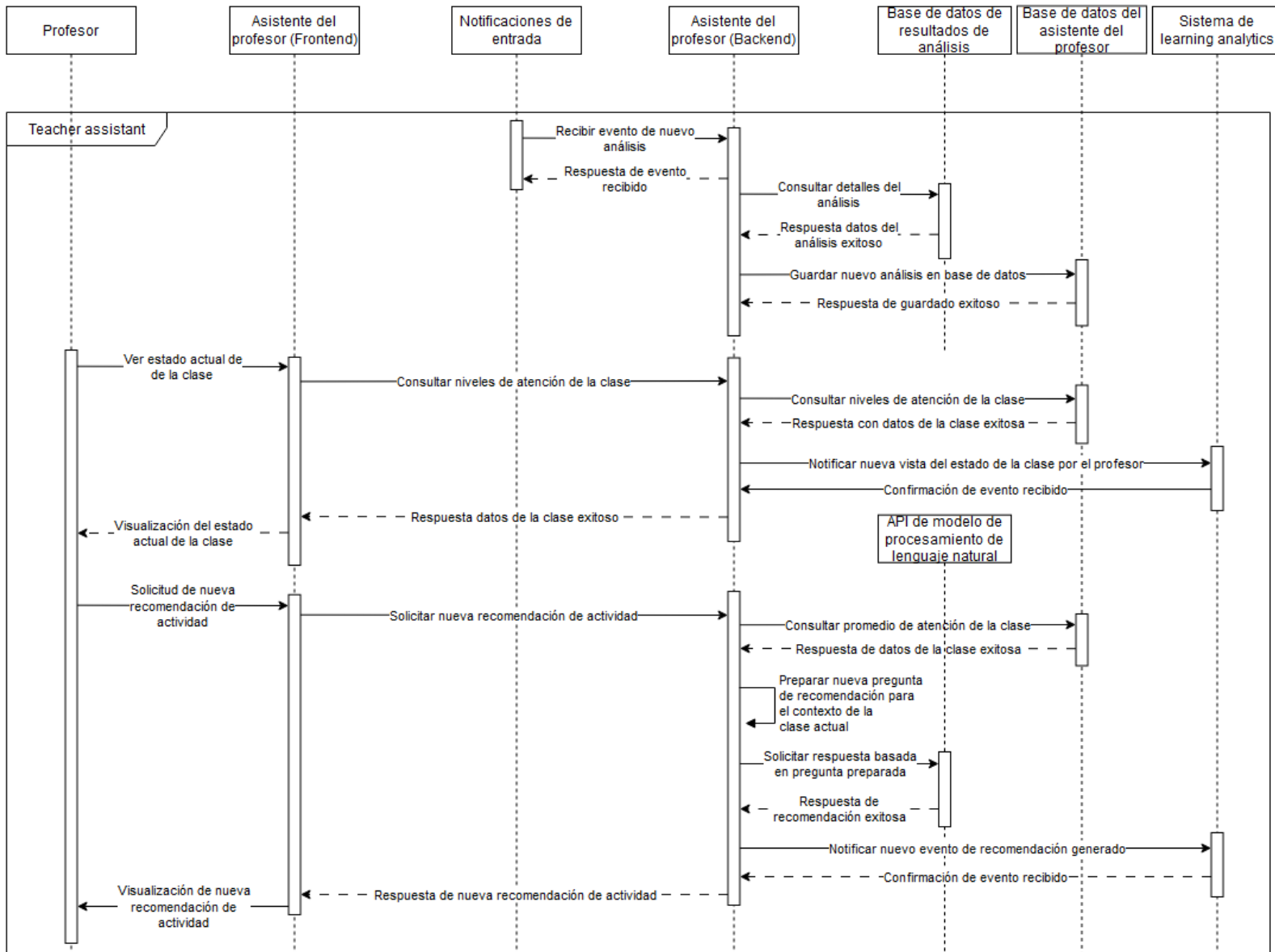


Ilustración 17: Diagrama de secuencia del sistema "Asistente del profesor" o "Teacher assistant". Creación propia

El sistema "Asistente del profesor", o "Teacher assistant" en inglés, es un sistema compuesto por una parte visual (Frontend), una parte de servidor (Backend) y una base de datos de asistente del profesor. También tiene interacciones con otros sistemas externos y otros sistemas de la solución. Posee una entrada asíncrona "Notificaciones de entrada", que se trata de una cola de mensajes la cual dispara una ejecución del "Asistente de Profesor Backend" que procesa el mensaje. Este mensaje es producido por el "Analizador de capturas de pantalla" y trae información sobre un nuevo análisis generado a partir de una captura de pantalla para una clase. El "Asistente del profesor Backend" usa esta información para realizar una consulta

a la base de datos de los resultados de los análisis y extraer más detalles sobre el análisis. Luego, estructura los datos de manera que puedan ser consultados por el frente visual (Frontend) después. Por otro lado, el Profesor interactúa con la aplicación visual de asistente del profesor (Frontend) de manera síncrona. Las operaciones disponibles para el profesor son “Ver estado actual de la clase” o “Solicitar nueva recomendación de actividad de aprendizaje activo”. Para suplir dichas peticiones, se usa la información guardada previamente por los eventos asíncronos que se van procesando a medida que se generan y completan nuevos análisis de imágenes de capturas de pantalla de los alumnos de la clase. Por último, las acciones del profesor generan eventos de analítica que se capturan a través de una cola de mensajes propia de otro sistema de la solución (“Learning analytics”).

- **Notificaciones de entrada:** Se refiere a una cola que recibe mensajes asíncronos generados por el “Analizador de capturas de pantalla”. Estos eventos se procesan, también, de manera asíncrona.
- **Asistente del profesor (Backend):** El “Backend” del asistente del profesor se encarga de consultar los detalles de cada nuevo análisis reportado en la base de datos de resultados de análisis, estructurar y guardar datos para que sean consultados después. También se encarga de procesar solicitudes de consulta de niveles de atención, estado de la clase y recomendaciones de actividad de aprendizaje activo. Para suplir estas peticiones síncronas, se consulta la base de datos del asistente del profesor y/o el modelo de procesamiento de lenguaje natural.
- **Base de datos de resultados de análisis:** En esta base de datos se almacenan los resultados de análisis de nivel de atención y el asistente del profesor consulta los detalles del análisis para poder preparar y guardar sus propios datos para su caso de uso en particular.
- **Base de datos del asistente del profesor:** Se encarga de guardar los datos necesarios para servir el estado actual de la clase en curso y los niveles de atención relevantes.
- **Sistema de learning analytics:** Este sistema se encarga de recibir los eventos relacionados con acciones del profesor y procesarlos. Para efectos de este diagrama, solo se muestra la confirmación de evento recibido, ya que es un evento asíncrono.
- **Profesor:** Ingresa a la aplicación y puede ver el estado actual de la clase, así como también solicitar una nueva recomendación de actividad de aprendizaje activo cuando lo desee.
- **Asistente del profesor (Frontend):** Es la parte visual del sistema, y permite al profesor realizar las acciones “ver el estado actual de la clase”, el cual también muestra los últimos análisis y rostros con su respectivo nivel de atención. También permite “solicitar nueva recomendación de actividad de aprendizaje activo”, y permite visualizar dicha actividad de manera visual para adaptar su proceso pedagógico.
- **API de modelo de procesamiento de lenguaje natural:** Es una llamada a un servicio externo que permite generar una respuesta en lenguaje natural,

la que simula una conversación con un humano experto. Se debe entregar una frase, o “prompt”, el cual puede ser construido a la medida. El servicio externo responde con una recomendación de actividad de acuerdo a la frase o “prompt” entregada.

6.2.4. Learning analytics

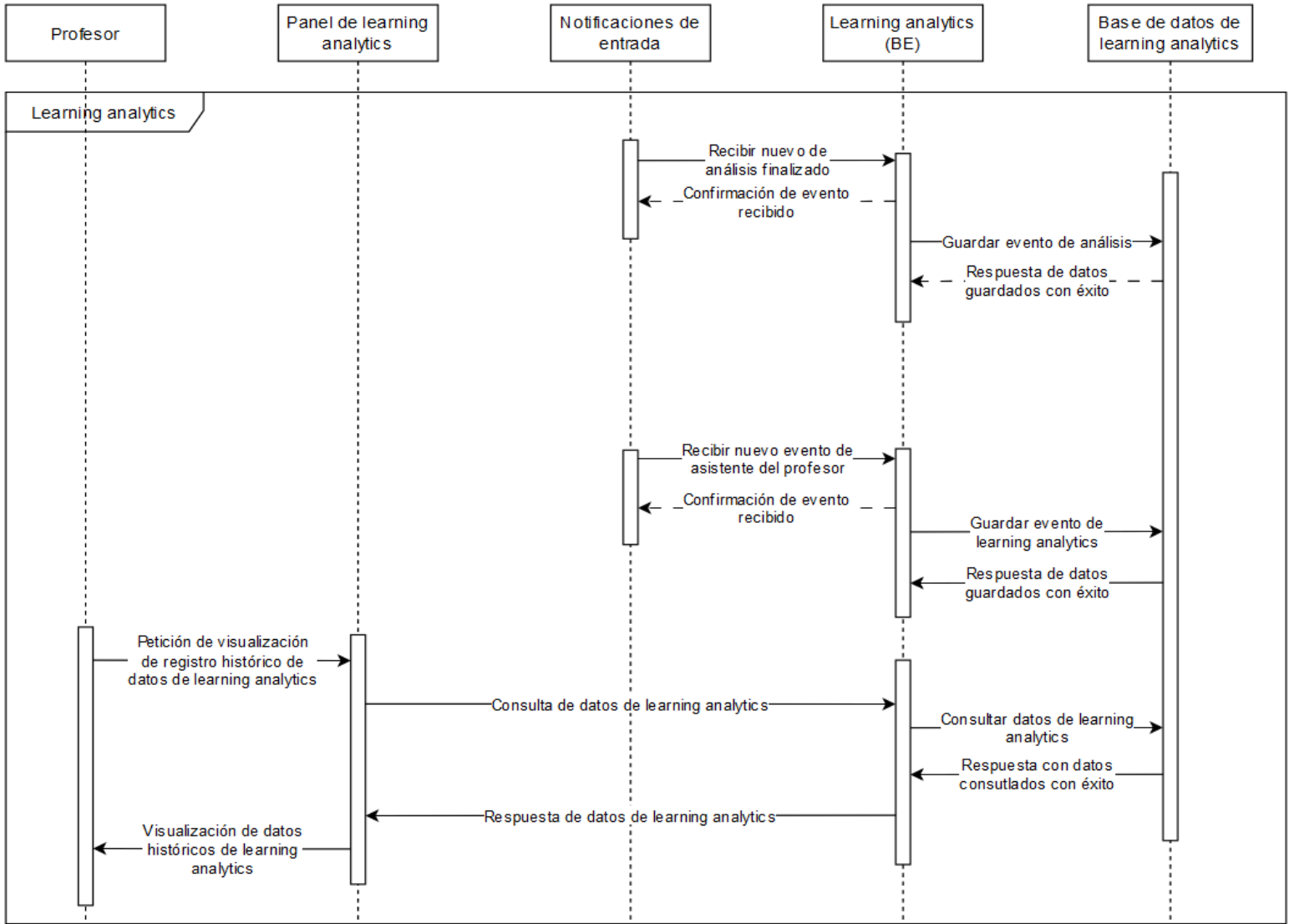


Ilustración 18: Diagrama de secuencia del sistema "Learning Analytics". Creación propia

El sistema de “Learning Analytics” es un módulo que permite guardar un histórico de datos de las clases que se toman y miden a lo largo de una clase virtual. Recibe tres tipos de eventos:

- Nuevo análisis generado.
- Profesor comenzó a medir nivel de atención.
- Profesor solicitó una nueva recomendación.

También tiene la capacidad de mostrar, de manera visual, los datos de las clases en un panel visual (Frontend).

- **Notificaciones de entrada:** Se trata de una cola que recibe mensajes de eventos. En este caso, se divide en dos tipos de eventos: eventos de “Nuevo análisis generado” y eventos del profesor.
- **Learning Analytics BE (Backend):** El lado del servidor del sistema cumple varias funciones: procesa los eventos de “nuevo análisis generado” y eventos del profesor. Ambos tienen lógicas un poco diferentes y por ende sus unidades de cómputo se muestran separadas en el diagrama. Al procesar estos eventos, se estructura y guarda la información en la base de datos de learning analytics. También se encarga de suplir los datos necesarios solicitados para mostrar en el panel visual de learning analytics, para lo cual debe consultar los datos de learning analytics.
- **Base de datos de learning analytics:** La base de datos de learning analytics recibe los datos generados en los eventos y los guarda. También recibe peticiones de datos del “Backend” sobre histórico de datos de clases.
- **Profesor:** Este es un actor que interactúa con el panel visual de learning analytics. Puede solicitar una visualización de registro histórico de datos de learning analytics.
- **Panel de learning analytics:** El panel de learning analytics se encarga de mostrar toda la parte visual necesaria para que el profesor pueda analizar y evaluar los diferentes datos de las diferentes clases o alumnos.

En la siguiente tabla se puede apreciar los sistemas y sub-sistemas de toda la solución. “Diagrama Padre” se refiere al diagrama que contiene el elemento en cuestión.

Para el “Tipo”, se usa la siguiente convención:

- **Actor:** se refiere a los actores (Personas) que interactúan con los sistemas.
- **Interno:** Se refiere a un servicio, unidad de cómputo o almacenamiento mantenido por la solución.
- **Externo (Nube):** Se refiere a servicios externos proporcionados por el proveedor de nube externo.
- **Externo (API):** Se refiere a servicios externos proporcionados por un tercero que no es un servicio de nube.

Nombre	Tipo	Diagrama Padre
Participante de captura de pantalla	Actor	Analizador de capturas de pantalla
Aplicación de captura de pantalla	Interno	Analizador de capturas de pantalla
Almacenamiento en la nube	Externo (Nube)	Analizador de capturas de pantalla
Analizador de nivel de atención	Interno	Analizador de capturas de pantalla
Base de datos de resultados de análisis	Interno	Analizador de capturas de pantalla
AWS Rekognition	Externo (Nube)	Analizador de capturas de pantalla
Servicio de notificaciones	Externo (Nube)	Analizador de capturas de pantalla
Profesor	Actor	Asistente del profesor
Asistente del profesor	Interno	Asistente del profesor
Notificaciones de entrada	Externo (Nube)	Asistente del profesor
Base de datos del asistente del profesor	Interno	Asistente del profesor
API de modelo de procesamiento de lenguaje natural	Externo (API)	Asistente del profesor
Panel de learning analytics	Interno	Learning analytics
Notificaciones de entrada	Externo (Nube)	Learning analytics
Learning analytics Backend	Interno	Learning analytics
Base de datos de learning analytics	Interno	Learning analytics

Tabla 2: Sistemas y aplicaciones que componen la arquitectura de la solución. Creación propia

6.3. DISEÑO DETALLADO

Para implementar el prototipo de la aplicación basado en la arquitectura de la solución se deben afianzar las tecnologías a usar a través de un diseño detallado de la solución. Este diseño detallado debe acomodarse a las decisiones tomadas en el diseño de la arquitectura y cumplir los requisitos expuestos. Para esto, se procede a realizar una elección de tecnologías y herramientas necesarias para construir el prototipo.

En primer lugar, se listan las técnicas, patrones o herramientas que se observan definidas desde el diseño de la arquitectura, las cuales guiarán la elección de otras herramientas o tecnologías para el diseño detallado de la solución:

- Sistemas desacoplados y microservicios
 - Los sistemas expuestos en el diseño de la arquitectura se encuentran desacoplados y desplegados de manera individual.
 - La notificación de los resultados de análisis se da por medio del patrón pub/sub (Reselman, 2021).
- Alojamiento en la nube pública
 - Se definió el uso de proveedores de nube pública sobre mantener servidores locales.
 - Existe preferencia en el uso de servicios de nube para suplir la demanda de la solución en cuanto a: alojamiento, despliegues, almacenamiento y registro.
- Uso de AWS Rekognition
 - De acuerdo con (Tabassum, Allen, & De, 2020), la herramienta que usaron para extraer el nivel de atención a partir de las emociones identificadas fue “AWS Rekognition”. Si se usará otro sistema para este fin, se podrían generar inconsistencias en los resultados e imprecisiones.
 - Se definió el uso del servicio “AWS Rekognition” (Amazon Web Services, 2023) para realizar los análisis e identificación de rostros y emociones.

6.3.1. Elecciones de tecnologías y herramientas

En esta sección se da un razonamiento acerca de las herramientas y tecnologías seleccionadas para realizar la implementación del prototipo, de acuerdo con la arquitectura diseñada.

En la actualidad existen muchos proveedores de nube públicas con algunas características similares como alta disponibilidad, centros de datos regionales distribuidos a lo largo del mundo, modelo de pago por uso, seguridad, entre otras.

Los proveedores líderes en la actualidad son “Amazon Web Services”, “Microsoft Azure” y “Google Cloud Platform”.

Dado que la técnica para calcular el nivel de atención de estudiantes se da usando la identificación de rostros y sus respectivas emociones usando “AWS Rekognition”, y de acuerdo al Cuadrante de Gartner 2022 (Bala, Smith, Ji, Wright, & Borrega, 2022), se encuentra que la elección de proveedor de nube más adecuada para el proyecto es “Amazon Web Services”, debido a que es el proveedor con mayor ventaja con respecto a los otros dos líderes: Esto hace que se encuentre una mayor base de usuarios y por tanto una comunidad más grande; lo que facilita la resolución de problemas, implementación de buenas prácticas, entre otras.

El hecho de usar “AWS Rekognition” para capturar las emociones y rostros de las imágenes refuerza el hecho de escoger AWS como proveedor de nube, ya que, al ser del mismo proveedor, los demás servicios se integrarán con mayor facilidad y los datos estarán alojados dentro del mismo proveedor.

Con base en esto, se finaliza “Amazon Web Services” como la elección de proveedor de nube pública.

Dentro del ecosistema de servicios y arquitecturas que ofrece AWS, destacan las arquitecturas y servicios manejados, o “serverless”. Las aplicaciones serverless son “Un modelo de desarrollo nativo de la nube que permite a los desarrolladores construir y correr aplicaciones sin tener que manejar servidores” (Red Hat, 2022). En particular, resalta el modelo computacional de Funciones como servicio, o “Functions-as-a-Service” (FaaS) por su nombre en inglés. Dicho modelo computacional delega la responsabilidad de administrar la plataforma sobre la que se ejecuta el código de la aplicación, permitiendo la creación de “funciones” que ejecutan cierta lógica de negocio propia del sistema. Como se expuso en el marco teórico, y lo expuesto por (Fee, 2020), (Red Hat, 2022), (Lassa, 2019) y (Mezzalira, Hyatt, Denti, & Jaupaj, 2022), se identifican las siguientes ventajas y compromisos al usar arquitecturas serverless:

Ventajas	Descripción
Escalamiento automático	Los servicios serverless en AWS escalan automáticamente para satisfacer la demanda. No hay que preocuparse por la gestión del escalamiento en la infraestructura
No hay manejo o administración de servidores	Se elimina la necesidad de aprovisionar y mantener servidores. Esto agiliza la entrega del software de manera considerable

Pago por uso	En la unidad de computo principal de serverless en AWS, AWS Lambda (Amazon Web Services, 2023), el modelo de pago es solo por el tiempo en que se use el servicio, por milisegundo. Otras tecnologías serverless en AWS tienen modelos de pago muy similares.
Implementación ágil	El uso de tecnologías serverless en AWS permite una rápida conexión e integración con otros servicios serverless en AWS, lo que agiliza considerablemente el desarrollo de las características de software
Alta disponibilidad	De manera automática, AWS realiza operaciones de replicación y redundancia para que las aplicaciones serverless estén disponibles cuando se necesiten, además de garantizar Acuerdos de Nivel de Servicio (SLA por sus siglas en inglés) para sus servicios serverless de entre 99.95% y 99.99% para la mayoría de sus servicios. (Amazon Web Services, 2023)

Tabla 3: Ventajas de usar tecnologías serverless para crear sistemas. Creación propia. Información tomada de (Fee, 2020), (Red Hat, 2022), (Lassa, 2019) y (Mezzalira, Hyatt, Denti, & Jaupaj, 2022)

Compromisos	Descripción
Tiempo de inicio en frío (Cold starts)	Las funciones serverless pueden experimentar un tiempo de “arranque en frío”, el cual introduce una latencia en la primera invocación debido al aprovisionamiento de la unidad de computo (Beswick, 2021)
Limitación en tiempo de ejecución	AWS Lambda tiene un tiempo de ejecución máximo de 15 minutos. Si se desea un tiempo de procesamiento mayor, deben buscarse alternativas dentro del diseño que suplan esa necesidad
Observabilidad	La observabilidad en una arquitectura serverless puede introducir complejidad adicional a la arquitectura debido a su naturaleza efímera. Existen herramientas para dar una visión completa que debe usarse si se quiere suplir este aspecto

Bloqueo de proveedor (vendor lock-in)	El uso de arquitecturas serverless en AWS hace que se cree un acoplamiento con el proveedor de nube pública (AWS). El esfuerzo necesario para migrar los sistemas es considerable y puede requerir una reescritura de código no trivial
---------------------------------------	---

Tabla 4: Desventajas de usar tecnologías serverless para crear sistemas. Creación propia. Información tomada de (Fee, 2020), (Red Hat, 2022), (Lassa, 2019) y (Mezzalana, Hyatt, Denti, & Jaupaj, 2022)

En cuanto a los compromisos, se definen las siguientes suposiciones:

- Los “arranques en frío” o “cold starts” por su nombre en inglés, se definen cuando se invoca por primera vez después de un tiempo.
 - Al haber hecho esta inicialización, la unidad de computo se mantiene inicializada por “un período no determinístico” (Beswick, 2021) de tiempo. Las invocaciones durante este período se conocen como arranques en caliente, o “warm starts”, los cuales no sufren de esta latencia adicional inherente de la primera invocación.
 - Los tiempos de latencia de “arranque en frío” dependen de varios factores como lo son (Solanki, 2021).
 - Tamaño de la función a ejecutar.
 - Muchas dependencias del código de las funciones.
 - Lenguajes de programación.
 - Configuración de bajos valores para la memoria y poder de procesamiento de la función.
 - Los tiempos de arranque en frío oscilan, en los peores casos, entre uno o dos segundos (Cui, 2020) (De Rooms, 2020), se determina que los arranques en frío tienen un impacto notable cuando se trata de llamados síncronos. Para las llamadas asíncronas en la solución que se está diseñando, este tiempo no es importante ya que ocurre solo durante la primera invocación, y teniendo en cuenta que el sistema que toma capturas de pantalla lo hace de manera cíclica, se descarta como un compromiso que impacte de manera negativa para los procesos asíncronos.
 - Para los procesos síncronos, existen diferentes técnicas para mitigar los arranques en frío, en específico, se tienen las siguientes consideraciones (Cui, 2020) (De Rooms, 2020).
 - Mantener el tamaño y las dependencias de la función al mínimo
 - Usar lenguajes de programación con mejores tiempos de arranque (Solanki, 2021).
 - Usar valores óptimos de memoria y poder de procesamiento para las funciones.

- Usar técnicas de mantenimiento de funciones en caliente, como llamados cíclicos o técnica de “conurrencia provisionada” para suplir la demanda.
- Para la limitante de tiempo de ejecución de 15 minutos, teniendo en cuenta los casos de uso y la arquitectura de la aplicación se identifica que ningún proceso debería estar cerca a este tiempo de ejecución, por lo que no es una limitante real para el diseño detallado de esta solución en particular.
- En cuanto a la observabilidad de los flujos de datos y ejecuciones, se debe garantizar una visibilidad completa de las ejecuciones serverless para tener la capacidad de realizar análisis de posibles ejecuciones lentas o llamados externos con mucha latencia.
 - “AWS XRay” es un servicio diseñado para la instrumentación de la solución. Al hacer uso de un kit de desarrollo de software, se puede implementar la captura y seguimiento de segmentos y subsegmentos de las ejecuciones de manera automática teniendo una visión completa de principio a fin de un llamado (síncrono y/o asíncrono) (Amazon Web Services, 2023).
 - Para identificar posibles errores en la aplicación se hace uso de “AWS Cloudwatch”, el cual es un servicio que permite monitorear recursos en la nube. Puede usarse para recolectar y almacenar registros de la aplicación, ver métricas o establecer alarmas (Amazon Web Services, 2023).
- Para el compromiso del bloqueo de vendedor (o “vendor lock-in” por su nombre en inglés), se acepta teniendo en cuenta las siguientes consideraciones:
 - Para mitigar el problema, se recomienda el desarrollo del código de la aplicación usando técnicas de servicios específicos de dominio, controladores y adaptadores. Esto con el fin de separar la lógica de negocio importante de las herramientas específicas que la usan (Vernon, 2013). Por lo anterior, se encuentra un bajo acoplamiento entre la lógica de negocio y la implementación de herramientas, lo que facilita el re-uso de dicha lógica de negocio en una posible migración a otro proveedor de nube o data center propio.
 - También, el uso de una herramienta que pueda adaptarse a diferentes proveedores de nube puede resultar en un tiempo de refactorización menor si se desea cambiar de proveedor. Para esta función, se sugiere el uso de “Serverless Framework”, el cual es una herramienta de infraestructura como código (IaC) que requiere una configuración mínima para desplegar aplicaciones serverless usando diferentes servicios de nube. La ventaja que ofrece es que dicha configuración puede adaptarse a diferentes proveedores de nube realizando ajustes en la misma (Serverless Inc., 2023) (Microsoft, 2023).

	artefact size	p10	p25	p50	p75	p99
control	222 bytes	205	213	222	236	325
control (post deploy)	222 bytes	226	235	247	268	596
AWS SDK is bundled but not required	9.47MB	235	245	260	274	386
AWS SDK is bundled but not required (post deploy)	9.47MB	258	272	291	317	660
control with big assets	60.02MB	460	485	514	548	779
control with big assets (post deploy)	60.02MB	506	534	573	640	1392
require built-in AWS SDK	265 bytes	446	455	471	487	566
require built-in AWS SDK (post deploy)	265 bytes	454	474	505	535	719
require AWS SDK via Layer	265 bytes	469	495	514	534	653
require AWS SDK via Layer (post deploy)	265 bytes	502	522	553	582	791
require bundled AWS SDK	9.47MB	588	604	624	644	769
require bundled AWS SDK (post deploy)	9.47MB	613	634	669	703	1002

Ilustración 19: Pruebas de tiempos de arranque en frío para diferentes configuraciones. Tomado de (Cui, 2020)

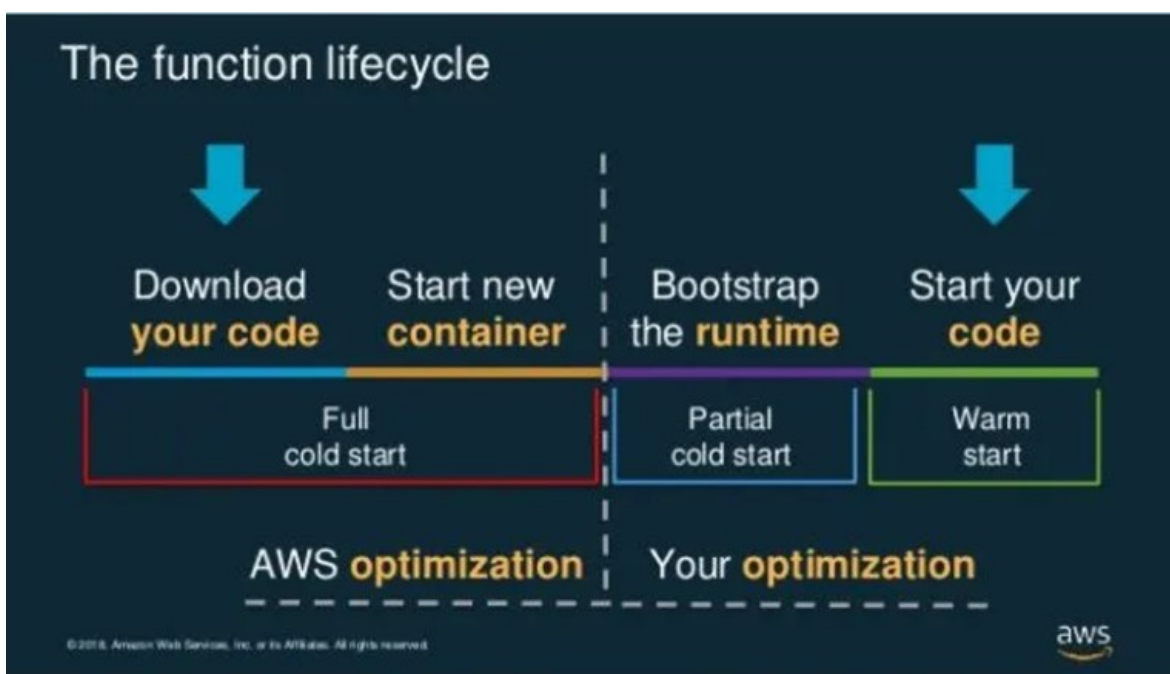


Ilustración 20: Ciclo de vida de una función AWS Lambda. Tomado de (Cui, 2020)

Teniendo en cuenta dichas consideraciones, se opta por una arquitectura serverless nativa de la nube, usando “Serverless Framework” (Serverless Inc., 2023) como herramienta de Infraestructura como Código, usando los servicios de nube ofrecidos por AWS para suplir las necesidades tecnológicas de la solución. A continuación, se ofrece un resumen de las tecnologías de seleccionadas junto con una breve descripción:

Ejecución de código como funciones (FaaS)

Se refiere a la ejecución del código de la aplicación, es decir, la lógica de negocio necesaria para cumplir su función, y demás necesidades de ejecución de código.

Nombre	Tipo	Descripción
AWS Lambda	Unidad de computo	AWS Lambda es una unidad de cómputo que aloja código y se ejecuta bajo demanda

Tabla 5: Elección de tecnología para ejecución de código como funciones (FaaS). Creación propia

Exponer funcionalidades a clientes (APIs)

Se requiere una compuerta de entrada que exponga funcionalidades síncronas a través de protocolos establecidos (REST).

Nombre	Tipo	Descripción
AWS API Gateway	Compuerta de entrada	API Gateway permite crear y mantener servicios con comunicación tipo REST y Web sockets a escala. Sirve como puerta de entrada para los llamados síncronos expuestos a diferentes clientes

Tabla 6:: Elección de tecnología para exponer funcionalidades a clientes (APIs). Creación propia

Almacenamiento en la nube

Se refiere al almacenamiento de diferentes tipos de datos en la nube, ya sean datos estructurados, relacionados o no relacionados, archivos, registros, entre otros.

Nombre	Tipo	Descripción
AWS DynamoDb	Base de datos	DynamoDb es una base de datos no relacional serverless. Tiene un diseño especializado en velocidad y escalabilidad
AWS RDS	Base de datos	RDS es un servicio maneja de base de datos relacional tipo SQL. Permite definir parámetros de auto escalamiento, replicación, entre otros
AWS CloudWatch	Observabilidad	CloudWatch es una herramienta de observabilidad completa que permite ver, almacenar y gestionar diferentes registros, métricas y alarmas
AWS Simple Queue Service (SQS)	Cola de Mensajería	SQS es una cola de mensajería usada para desacoplar diferentes procesos asíncronos. Permite recibir, almacenar y direccionar mensajes a consumidores
AWS S3	Almacenamiento de archivos	S3 es un servicio administrado de almacenamiento de archivos en la nube con diferentes características que permite

		almacenar archivos de manera durable, segura y rentable
--	--	---

Tabla 7: Elección de tecnología para almacenamiento de datos en la nube. Creación propia

Observabilidad

Las herramientas de observabilidad permiten tener una visión de lo que pasa en los sistemas que componen la aplicación, permitiendo diagnosticar, analizar posibles mejoras y anticiparse a posibles errores minimizando el impacto de dichos errores.

Nombre	Tipo	Descripción
AWS CloudWatch	Observabilidad	CloudWatch es una herramienta de observabilidad completa que permite ver, almacenar y gestionar diferentes registros, métricas y alarmas
AWS XRay	Instrumentación	XRay es un servicio de instrumentación que permite tener una visión completa de principio a fin en aplicaciones. Es especialmente útil en arquitecturas de microservicios y serverless

Tabla 8: Elección de tecnología para herramientas de Observabilidad. Creación propia

Seguridad

Los servicios enfocados en seguridad permiten una manera fácil y rápida de encriptar datos, ya sea encriptación de datos en tránsito o encriptación de datos en reposo. También proveen protecciones a los servicios de posibles ciberataques y manejo de acceso a otros recursos.

Nombre	Tipo	Descripción
AWS KMS	Encriptación	KMS es un servicio integrado en los demás servicios de AWS que tengan que ver con datos. Permite almacenar de manera simple y rápida datos encriptados. La llave de encriptación puede ser manejada por el proveedor de nube o por el cliente
AWS WAF	Cortafuegos (Firewall)	WAF es un servicio de AWS que provee protección como control de "bots", XSS, inyección de SQL, entre otros.
AWS IAM	Administración de accesos	IAM es un servicio que permite establecer acceso sobre los recursos de nube, permitiendo definir permisos para acceder a recursos de nube limitados

		solo a los permisos necesarios para ejecución. Esto lo logra a través del uso de Roles y Políticas que pueden adherirse a otros servicios
--	--	---

Tabla 9: Elección de tecnología para seguridad y manejo de acceso. Creación propia

Comunicación asíncrona entre servicios

Para notificaciones y servicios de mensajería que faciliten la comunicación entre diferentes servicios, así como comunicaciones asíncronas, se hace uso del patrón de cola de mensajes y pub/sub, bajo la modalidad de “Fan-Out” (Richardson & Fröhner, 2019), la cual es un patrón de integración de microservicios.

Nombre	Tipo	Descripción
AWS Simple Queue Service (SQS)	Cola de Mensajería	SQS es una cola de mensajería usada para desacoplar diferentes procesos asíncronos. Permite recibir, almacenar y direccionar mensajes a consumidores
AWS Simple Notification Service (SNS)	Servicio de Mensajería tipo “pub/sub”	SNS es un servicio de mensajería tipo “pub/sub” (publicador/suscriptor por su nombre en inglés). Se usa para notificar de manera asíncrona sobre un mensaje

Tabla 10: Elección de tecnología para comunicación basado en mensajes y entres servicios. Creación propia

Herramientas de integración y despliegues

En cuanto a herramientas de desarrollo y despliegues, se debe tener en cuenta donde estará alojado el código fuente y cómo se realizarán los despliegues.

Nombre	Tipo	Descripción
Git	Control de versión de código fuente	Git es una herramienta completa que permite tener versiones del código fuente de aplicaciones, permitiendo tener un motor de versionamiento para el código de las aplicaciones
Github	Control de versión de código fuente (Git) distribuido	Github es una plataforma que permite almacenar y mantener versionamiento del código usando “Git” a través de repositorios. Dichos repositorios se almacenan en Github, y permiten colaboración entre diferentes desarrolladores

AWS Code Build	Integración continua	CodeBuild es la herramienta nativa de AWS para hacer integración continua del código. Algunas de las prácticas comunes involucran compilar el código, verificar pruebas unitarias y generar artefactos para el despliegue. En general, ayuda a integrar el código y facilitar los despliegues (Rehkopf, 2023)
AWS Code Pipeline	Automatización de integración y despliegues	Code Pipeline es la herramienta dentro del conjunto de servicios para desarrolladores de AWS que permite automatizar la integración y entrega de software. Se pueden definir pasos a seguir siempre que se detecte un cambio en el código fuente, lo que genera una nueva integración y despliegue de la aplicación, minimizando la intervención manual
Serverless Framework	Marco de trabajo de infraestructura como código	Serverless es un marco de trabajo de infraestructura como código (IaC) que permite configurar, construir y desplegar aplicaciones en la nube (Serverless Inc., 2023).
AWS Cloudformation	Herramienta de infraestructura como código	Cloudformation es el servicio de Infraestructura como código (IaC) de AWS para crear, actualizar o eliminar recursos de nube.
AWS Amplify	Herramienta que permite integrar recursos de nube con interfaces de usuario	AWS Amplify es una herramienta de "Backend" que permite facilitar el uso de recursos y servicios de nube a aplicaciones de interfaz de usuario. Para el alcance de este prototipo, se usará el módulo de despliegue (Amazon Web Services, 2021)

Tabla 11: Elección de tecnología para herramientas de integración y despliegues. Creación propia

Tecnologías y herramientas para el desarrollo

Define el lenguaje de programación, entorno de ejecución del código y editor de código para construir el prototipo.

Nombre	Tipo	Descripción
Javascript	Lenguaje de programación	Javascript es un lenguaje de programación interpretado y caracterizado por su sistema de tipado dinámico, ha ganado notoriedad en los

		últimos años y actualmente es considerado el lenguaje de programación más usado (Stack Overflow, 2023) (Jet Brains, 2022). Además de gozar de una comunidad activa y una evolución constante, es uno de los lenguajes que provee mejores desempeños para el computo en AWS Lambda (Solanki, 2021)
Typescript	Lenguaje de programación	Typescript es un lenguaje de programación que es un superconjunto de javascript, añadiendo características de programación estática como tipos, clases, interfaces, entre otros. Su objetivo es mejorar y dar confianza en los desarrollos que usen javascript. Dado que en la fase de compilación se transforma a javascript puro, puede correrse en cualquier plataforma que soporte javascript (Microsoft, 2023)
Node.js	Entorno de ejecución de servidor	Node.js es un entorno de ejecución multiplataforma y de código abierto, con un diseño asíncrono y orientado a eventos, permite manejar muchas conexiones simultáneamente sin bloquearse (OpenJS Foundation, 2023).
Visual Studio Code	Editor de código	Visual studio code (abreviado comúnmente como vscode) es un editor de texto especializado en construir y depurar aplicaciones modernas en la web y en la nube. Es gratis y tiene un gran número de extensiones que permiten mejorar la productividad de los desarrolladores (Microsoft Azure, 2023)

Tabla 12: Elección de tecnología para desarrolladores y construcción de código. Creación propia

6.3.2. Diagramas detallados

En cuanto a qué modelo usar para mostrar el diseño detallado de la solución, el modelo “C4” de arquitectura de software se basa en las abstracciones de un sistema de software, siendo un modelo basado en 4 niveles de visión de una arquitectura:

- Diagrama de Contexto (Nivel 1).
- Diagrama de Contenedores (Nivel 2).
- Diagrama de Componentes (Nivel 3).

- Diagrama de Código (Nivel 4).

Cada uno de estos niveles están relacionados entre sí de manera jerárquica. El primer nivel, “Diagrama de Contexto”, es un punto de partida que permite visualizar la solución sin detalles. Se centra en personas (actores, roles, personas, etc), y en cómo interactúan sistemas de software.

En el segundo nivel, “Diagrama de Contenedores”, se describe cómo encaja cada sistema descrito en el diagrama de “Contexto” en el ambiente de TI. Un “contenedor” se refiere a una unidad separada y desplegable que cumple una función como ejecutar código, almacenar datos, entre otros. Es útil para entender cómo se distribuyen las responsabilidades y muestra las principales elecciones de tecnologías.

En el tercer nivel, “Diagrama de Componentes”, se descompone cada contenedor y da a entender como un contenedor está compuesto por un número de componentes. Muestra dichos componentes, sus responsabilidades y da más detalles de implementación.

En el cuarto nivel, “Diagrama de Código”, se dan detalles de implementación del código que hace parte de cada componente encontrado en el “Diagrama de Componentes” usando lenguaje como UML, Diagramas entidad-relación o similares. Este es un nivel opcional y no se recomienda a excepción de componentes muy complejos.

El modelo C4 es efectivo para definir la arquitectura de una solución de software por sus características principales:

- **Abstracción jerárquica:** El hecho de desglosar la arquitectura en cuatro niveles jerárquicos permite visualizar diferentes niveles de detalle y tener una visión desde lo general hasta lo específico.
- **Comunicación clara:** El modelo “C4” busca generar un estándar para definir arquitecturas de software al proveer una estructura simple y fácil de entender, que ayuda a sincronizar a diferentes personas tanto técnicas como no-técnicas.
- **Inducción y documentación:** Los diagramas del modelo “C4” mejoran los procesos de inducción de nuevos empleados y sirven como documentación útil de una solución de software.

(Brown, 2023)

A continuación, se exponen los diferentes servicios que componen la arquitectura de la solución bajo el modelo C4, luego se presentan de manera visual junto con una explicación de cada elemento por cada diagrama. El nivel 4 (Diagrama de código) se omite acatando la recomendación del modelo de que puede ser generado bajo demanda usando herramientas automatizadas una vez el código esté finalizado:

Nombre	Tipo de diagrama	Diagrama padre
"Class Attention"	Contexto (Nivel 1)	-
"Screenshot Analyzer"	Contenedor (Nivel 2)	"Class Attention"
"Screenshot App"	Componente (Nivel 3)	"Screenshot Analyzer"
"Attention Level Analyzer"	Componente (Nivel 3)	"Screenshot Analyzer"
"Teacher Assistant"	Contenedor (Nivel 2)	"Class Attention"
"Teacher Feedback BE"	Componente (Nivel 3)	"Teacher Assistant"
"Teacher Feedback FE"	Componente (Nivel 3)	"Teacher Assistant"
"Learning Analytics"	Contenedor (Nivel 2)	"Class Attention"
"Learning Analytics Backend"	Componente (Nivel 3)	"Learning Analytics"
"Learning Analytics Dashboard"	Componente (Nivel 3)	"Learning Analytics"

Tabla 13: Jerarquía de diagramas dentro del diseño detallado bajo modelo C4. Creación propia

6.3.2.1. Diagrama de contexto “Class Attention”

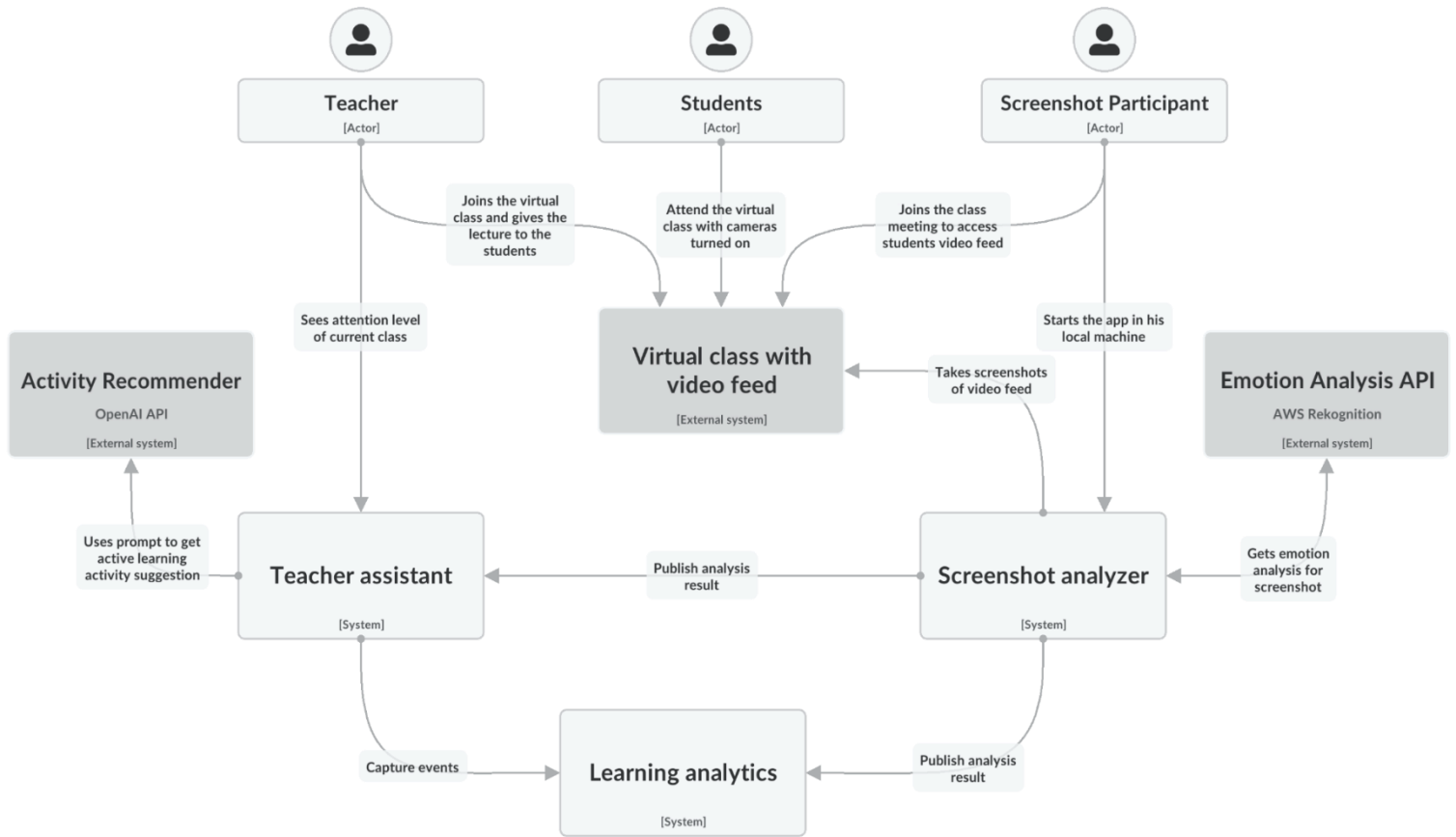


Ilustración 21: Diagrama de contexto de la solución. Creación propia

Diagrama de contexto “Class Attention”		
Elemento	Tipo	Descripción
Teacher (Profesor)	Actor	El profesor es quién dicta la clase virtual a sus alumnos a través de la plataforma “Microsoft Teams”
Students (Estudiantes)	Actor	Los estudiantes ingresan a la clase virtual con las cámaras de sus dispositivos encendidas y asegurándose de que dicho dispositivo pueda captar su rostro

Screenshot Participant (Participante de captura de pantalla)	Actor	El participante de captura de pantalla es una persona que ingresa a la misma clase virtual que el profesor y los estudiantes. Dicha persona debe poder configurar su "Microsoft Teams" para ver el video de los rostros de los estudiantes
Clase virtual con captura de video	Sistema externo	Se trata de la llamada virtual a la que se unen los demás actores: Profesor, Estudiantes y Participante de captura de pantalla. La plataforma es "Microsoft Teams" y los alumnos transmiten el video del rostro de ellos mismos a lo largo de la clase
"Screenshot Analyzer"	Sistema	El sistema "anizador de screenshot" Se encarga de tomar las capturas de pantalla del video de los rostros de los alumnos proveída por el "participante de captura de pantalla". Almacena y provee un análisis de cada rostro identificado. Por último, transmite los resultados a los demás sistemas suscritos
Analizador de emociones	Sistema Externo	El analizador de emociones es un servicio manejado que toma una imagen como entrada del proceso. Identifica cada rostro en la imagen y provee las emociones presentes en ese rostro junto con un coeficiente de confianza
"Teacher Assistant"	Sistema	El asistente del profesor es un sistema tipo aplicación web que permite al profesor visualizar el nivel de atención de los estudiantes de su clase, así como tener acceso a recomendaciones de actividades de aprendizaje activo
Recomendador de actividades	Sistema externo	El "recomendador" de actividades es un sistema externo que usa el modelo de procesamiento de lenguaje natural "GPT-4" para que, a través de su API, se le ingrese una frase o "prompt" que permita al modelo generar una actividad acorde al contexto de la clase: tema de la clase, nivel de estudios de los estudiantes, tiempo de clase y tiempo que se ha identificado desatención en los estudiantes
"Learning Analytics"	Sistema	El servicio de "learning analytics" es un servicio que captura determinados eventos de otros

		sistemas. Almacena estos eventos de manera histórica y permite visualizar estos datos a través de un panel visual
--	--	---

Tabla 14: Elementos dentro del diagrama de contexto bajo el modelo C4. Creación propia

6.3.2.2. Diagrama de contenedores: “Screenshot Analyzer”

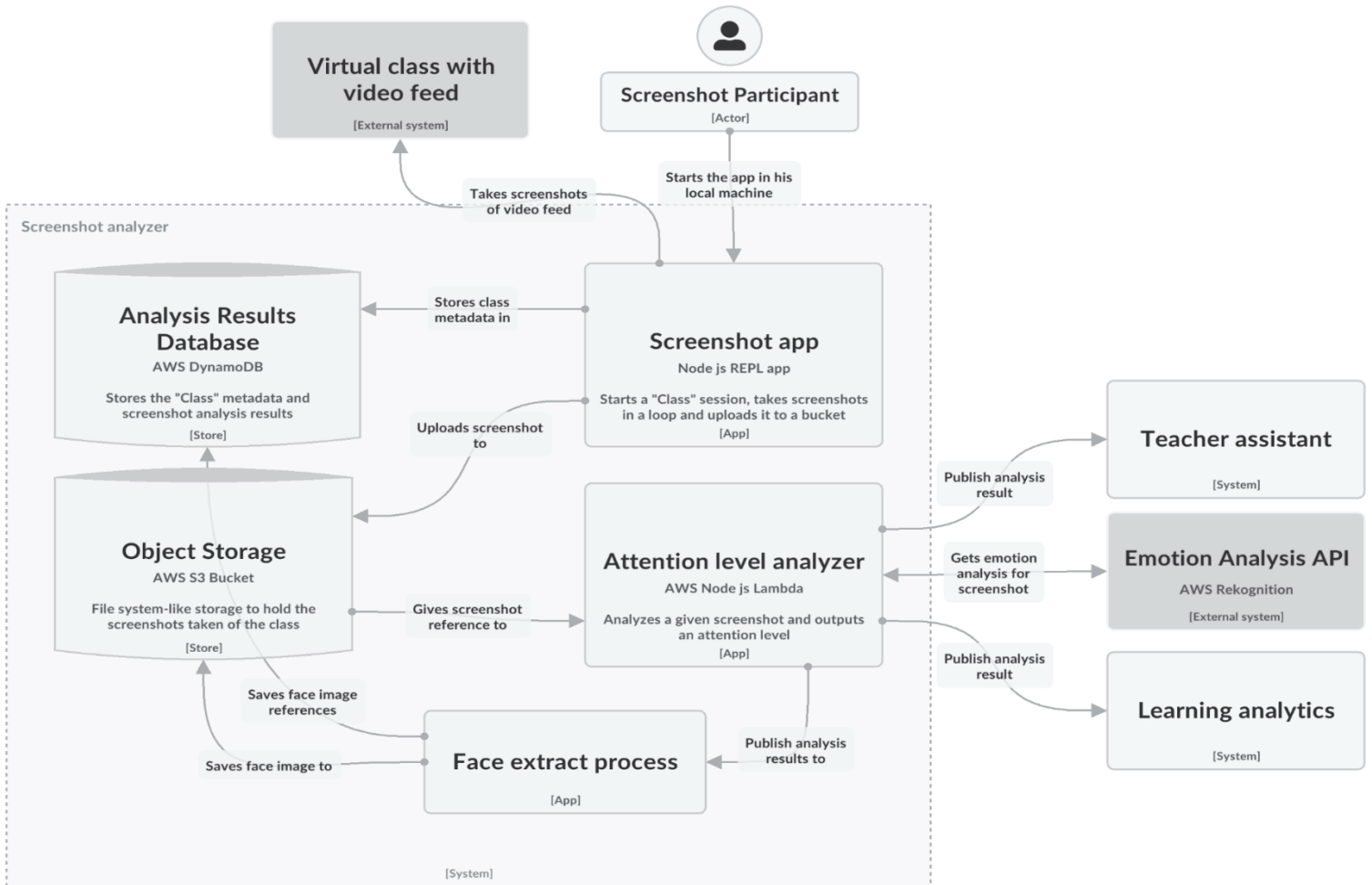


Ilustración 22: Diagrama de contenedores para analizador de capturas de pantalla. Creación propia

Diagrama de contenedores “Screenshot Analyzer”		
Elemento	Tipo	Descripción
“Screenshot app”	Contenedor - Aplicación Node.js REPL	La aplicación de captura de pantalla es una aplicación de consola que se encarga de tomar capturas de pantalla y almacenarlas en la nube. El participante de captura de pantalla es el actor que la usa y tiene la responsabilidad de iniciar la captura de imágenes de la clase
“Object Storage”	Contenedor - AWS S3 Bucket	Este es un almacenamiento de nube que permite guardar archivos en forma de “objetos”. Es el encargado de almacenar las capturas de pantalla y los rostros identificados en cada imagen
“Attention level analyzer”	Contenedor - Aplicación de servidor AWS Lambda Node.js	Esta unidad de cómputo tiene por objetivo analizar capturas de pantalla, extrayendo los rostros individuales, obteniendo coeficientes de emociones por cada rostro y asignándoles un nivel de atención en forma de porcentaje basado en las emociones identificadas. Luego, guarda y transmite los resultados a los clientes suscritos
Analysis Results Database	Contenedor - Base de datos AWS DynamoDB	La base de datos “Analysis results database” es una tabla de la base de datos no-relacional “dynamodb” que se encarga de guardar los resultados del análisis, así como identificador por rostro, nivel de atención por cada rostro y por la clase en general, y también almacena otros datos sobre cuando comenzó la clase

Tabla 15: Elementos dentro del diagrama de contenedores del sistema “Screenshot Analyzer”, o Analizador de Captura de Pantalla bajo el modelo C4. Creación propia

		guardar o leer información
“AWS”	Componente	El módulo AWS es un agrupamiento de funciones que permiten usar los diferentes servicios de nube. Implementa el kit de desarrollo de software (SDK) oficial de AWS
“Lib”	Componente	Lib se trata de un grupo de interfaces creadas para interactuar con recursos externos
“Config”	Componente	El módulo de configuración contiene los diferentes valores que la aplicación requiere para cumplir sus funciones
“Utilities”	Componente	El módulo de Utilidades es un agrupamiento de métodos reusables que permiten a los servicios definidos realizar diferentes funciones

Tabla 16: Elementos dentro del diagrama de componentes del contenedor "Attention level analyzer", o Analizador de nivel de atención, bajo el modelo C4. Creación propia

6.3.2.4. Diagrama de componentes “Screenshot app”

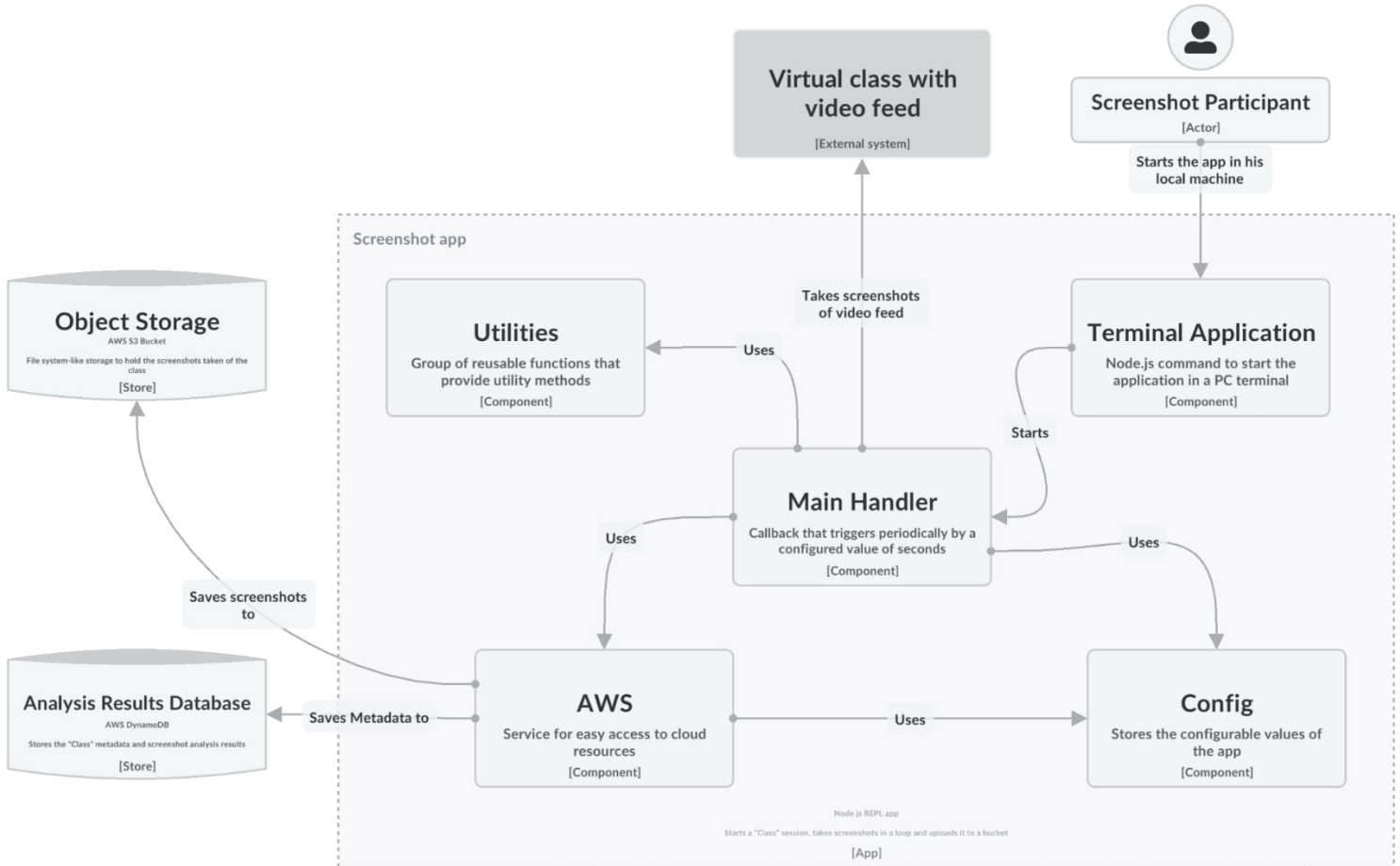


Ilustración 24: Diagrama de componentes para la aplicación de captura de pantalla. Creación propia

Diagrama de componentes “Screenshot App”		
Elemento	Tipo	Descripción
“Terminal Application”	Componente	La aplicación de terminal se refiere al comando necesario para iniciar el proceso de captura de pantallas y es el punto de partida del proceso
“Main Handler”	Componente	Es una función que se llama recursivamente en un intervalo de tiempo determinado y configurable. Dentro de cada ciclo de ejecución,

		se encuentran los diferentes métodos para tomar captura de pantalla y cargar al almacenamiento de nube
“Utilities”	Componente	Es un agrupamiento de métodos reusables que permiten realizar diferentes funciones
“AWS”	Componente	El módulo de “AWS” es un agrupamiento de funciones que permiten acceder y leer o escribir. Implementa el kit de desarrollo de software (SDK) oficial de AWS
Config	Componente	El módulo de configuración contiene los diferentes valores que la aplicación requiere para cumplir sus funciones

Tabla 17: Elementos dentro del diagrama de componentes del contenedor "Screenshot app", o aplicación de captura de pantalla, bajo el modelo C4. Creación propia

6.3.2.5. Diagrama de contenedores “Teacher Assistant”

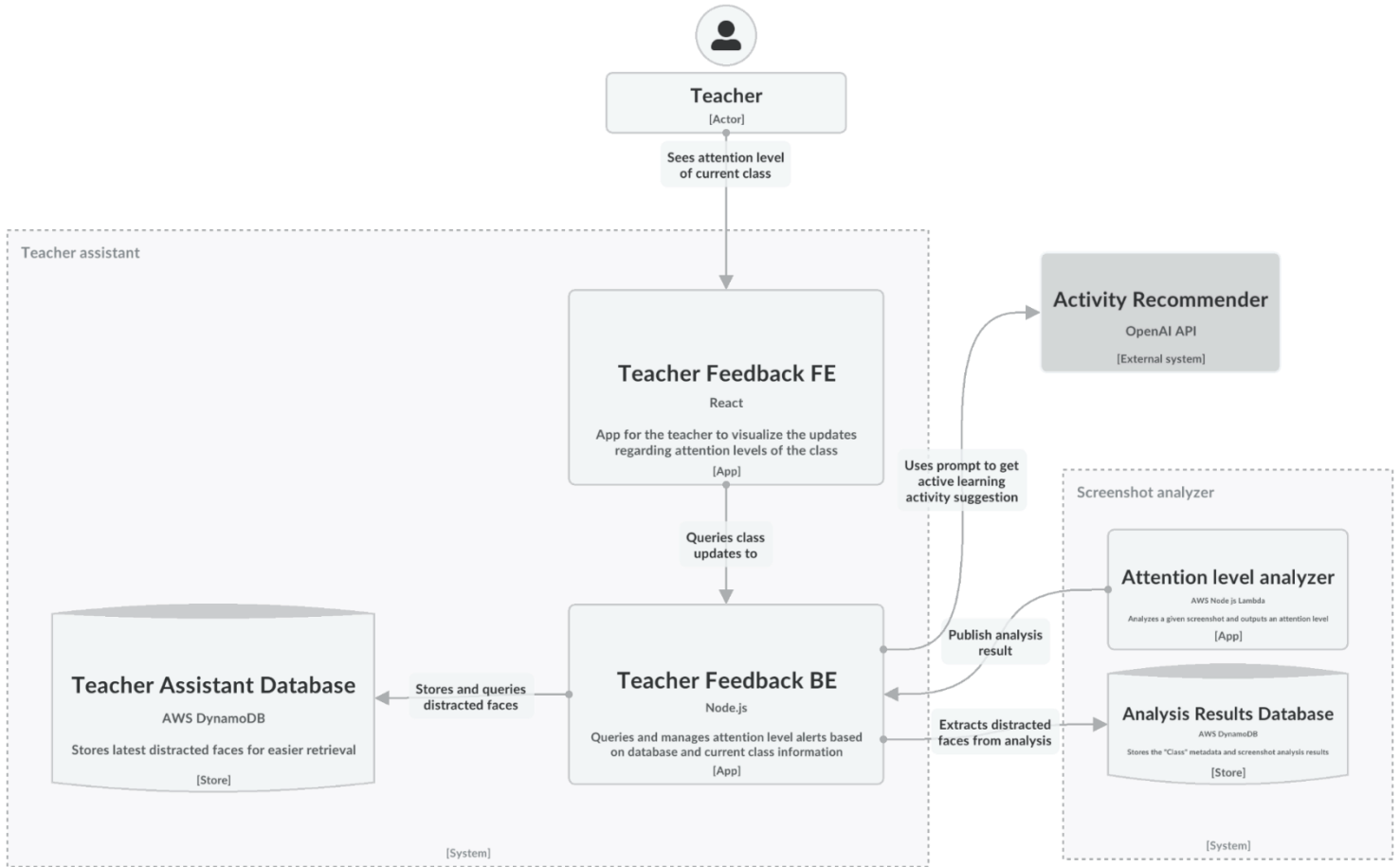


Ilustración 25: Diagrama de contenedores para el asistente del profesor. Creación propia

Diagrama de contenedores “Teacher Assistant”		
Elemento	Tipo	Descripción
“Teacher Feedback FE”	Contenedor - Aplicación web de cliente React.js	La aplicación permite visualizar el nivel de atención de la clase actual, también puede pedir recomendaciones de actividades las cuales serán mostradas en la vista de la aplicación web
“Teacher	Contenedor -	Esta aplicación tiene la capacidad de procesar

Feedback BE”	Aplicación de servidor AWS Lambda Node.js	los resultados de análisis de capturas de imágenes y almacenar dichos datos para ser consultados por la aplicación web visual. También tiene la capacidad de realizar recomendaciones de actividades de aprendizaje activo con la ayuda de procesamiento de lenguaje natural
“Teacher Assistant Database”	Contenedor - Tabla de base de datos “AWS DynamoDB”	Se refiere a la tabla en la base de datos “aws dynamodb” donde se almacenarán los resultados de análisis, referencias a imágenes de rostros de estudiantes, y demás información para cumplir con sus funciones

Tabla 18: Elementos dentro del diagrama de contenedores "Teacher assistant", o Asistente del profesor, bajo el modelo C4. Creación propia

6.3.2.6. Diagrama de componentes “Teacher Feedback BE”

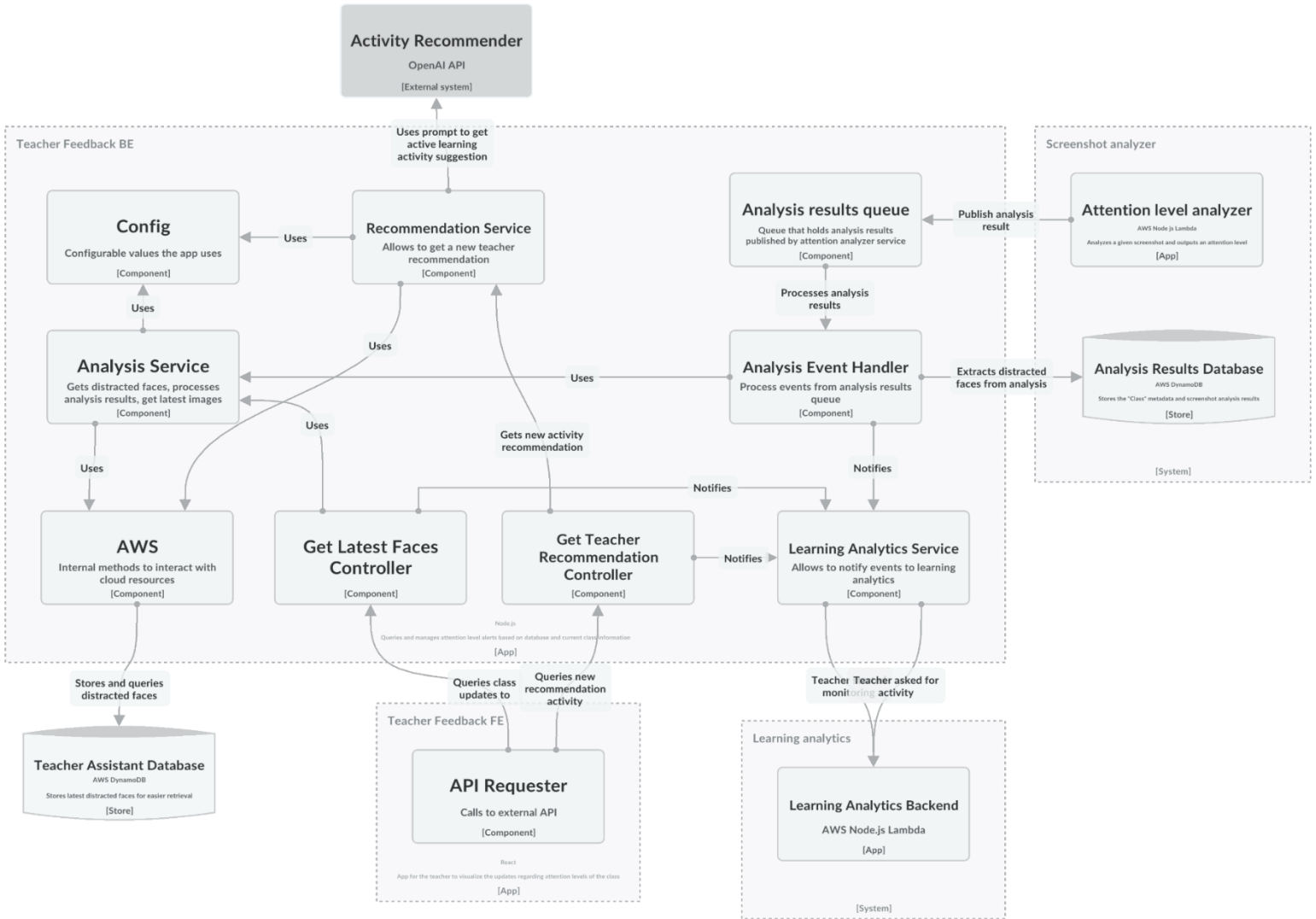


Ilustración 26: Diagrama de contenedores del backend para el asistente del profesor. Creación propia

Diagrama de componentes “Teacher Feedback BE”		
Elemento	Tipo	Descripción
“Analysis Results Queue”	Componente	Este es un componente tipo Cola de mensajes. La cola está suscrita a las comunicaciones que se envían sobre los resultados de análisis de imágenes.

“Analysis Event Handler”	Componente	El manejador de eventos de análisis se ejecuta por cada mensaje que se recibe de la cola de mensajes de resultados “Analysis Results Queue”. Se encarga de procesar dicho mensaje y notificar al servicio de “learning analytics”
“Analysis Service”	Componente	El servicio de análisis es un componente capaz de procesar los resultados de las imágenes, guardar dichos procesamientos, consultar rostros y nivel de atención relevantes para la visualización de datos
“Config”	Componente	El módulo de configuración contiene los valores configurables que la aplicación usa para cumplir sus funciones
“Recommendation Service”	Componente	El servicio de recomendaciones es un servicio que permite generar una nueva recomendación basado en el contexto de la clase y el nivel de atención
“AWS”	Componente	El módulo “AWS” es un agrupamiento de funciones que permiten interactuar con los diferentes servicios de nube de AWS a través del kit de desarrollo de software (SDK) oficial
“Get Latest Faces Controller”	Componente	El controlador de últimos rostros se encarga de recibir el llamado de la aplicación web “Teacher Feedback FE”, realizar unas validaciones iniciales y usa el servicio de análisis para servir la solicitud
“Get Teacher Recommendation Controller”	Componente	El controlador de recomendaciones se encarga de recibir y validar el llamado de la aplicación web “Teacher Feedback FE”,
“Learning Analytics Service”	Componente	El servicio de “Learning Analytics” se encarga de notificar eventos que ocurran dentro del sistema para poder capturar un registro histórico de los diferentes eventos

Tabla 19: Elementos dentro del diagrama de componentes del contenedor “Teacher feedback backend”, o Backend del asistente del profesor, bajo el modelo C4. Creación propia

6.3.2.7. Diagrama de componentes “Teacher Feedback FE”

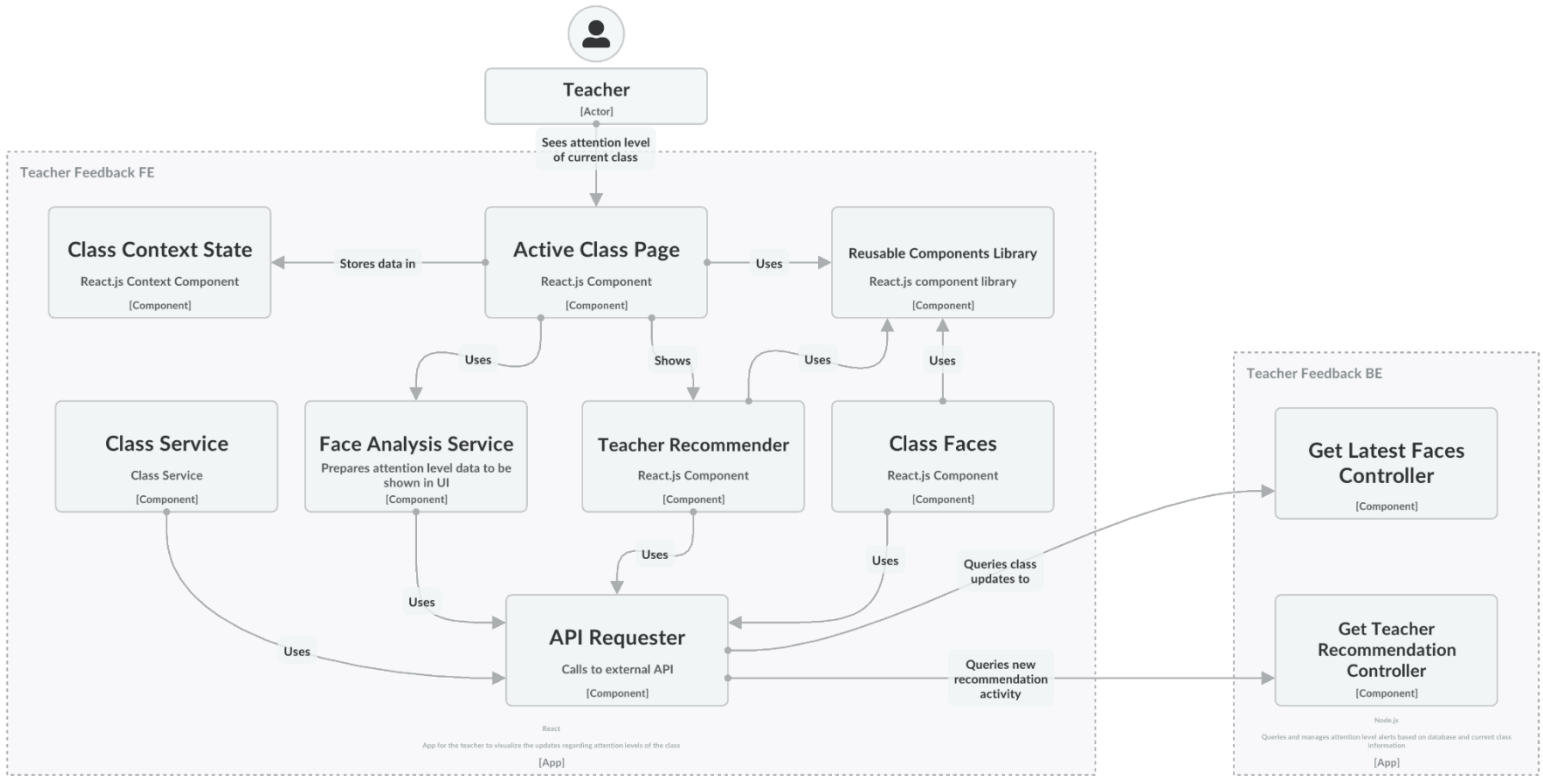


Ilustración 27: Diagrama de componentes para el frontend del asistente del profesor. Creación propia

Diagrama de componentes “Teacher Feedback FE”		
Elemento	Tipo	Descripción
“Active Class Page”	Componente	La página de la clase activa es la vista que muestra el estado de la clase actual junto con la información del nivel de atención de los alumnos y las recomendaciones de actividades
“Class Context State”	Componente	El contexto de la clase se refiere al conjunto de datos necesarios que deben almacenarse y mantenerse para que las vistas puedan mostrar la información de manera visual.
“Reusable Components”	Componente	La librería de componentes reusables es un conjunto de elementos visuales con lógica que

Library”		cumplen funciones específicas. Se pueden importar y re-usar por otros elementos dentro de los componentes visuales
“Class Service”	Componente	El servicio de la clase hace referencia a la lógica del negocio para consultar todos los datos necesarios que tienen que ver con la clase en general
“Face Analysis Service”	Componente	El servicio de análisis de rostros es el servicio que implementa la lógica necesaria para consultar y estructurar los datos sobre el nivel de atención de los estudiantes
“Teacher Recommender”	Componente	El “recomendador” del profesor es un módulo que permite visualizar las recomendaciones de actividades y permite también solicitar nuevas recomendaciones
“Class Faces”	Componente	Se refiere a un módulo capaz de mostrar los rostros de los alumnos que participan de la clase actual, junto con su respectivo nivel de atención
“API Requester”	Componente	Este se refiere a funciones reusables que facilitan las peticiones a los sistemas externos a la aplicación web de cliente

Tabla 20: Elementos dentro del diagrama de componentes del contenedor "Teacher feedback frontend", o interfaz de usuario del asistente del profesor, bajo el modelo C4. Creación propia

6.3.2.8. Diagrama de contenedores “Learning analytics”

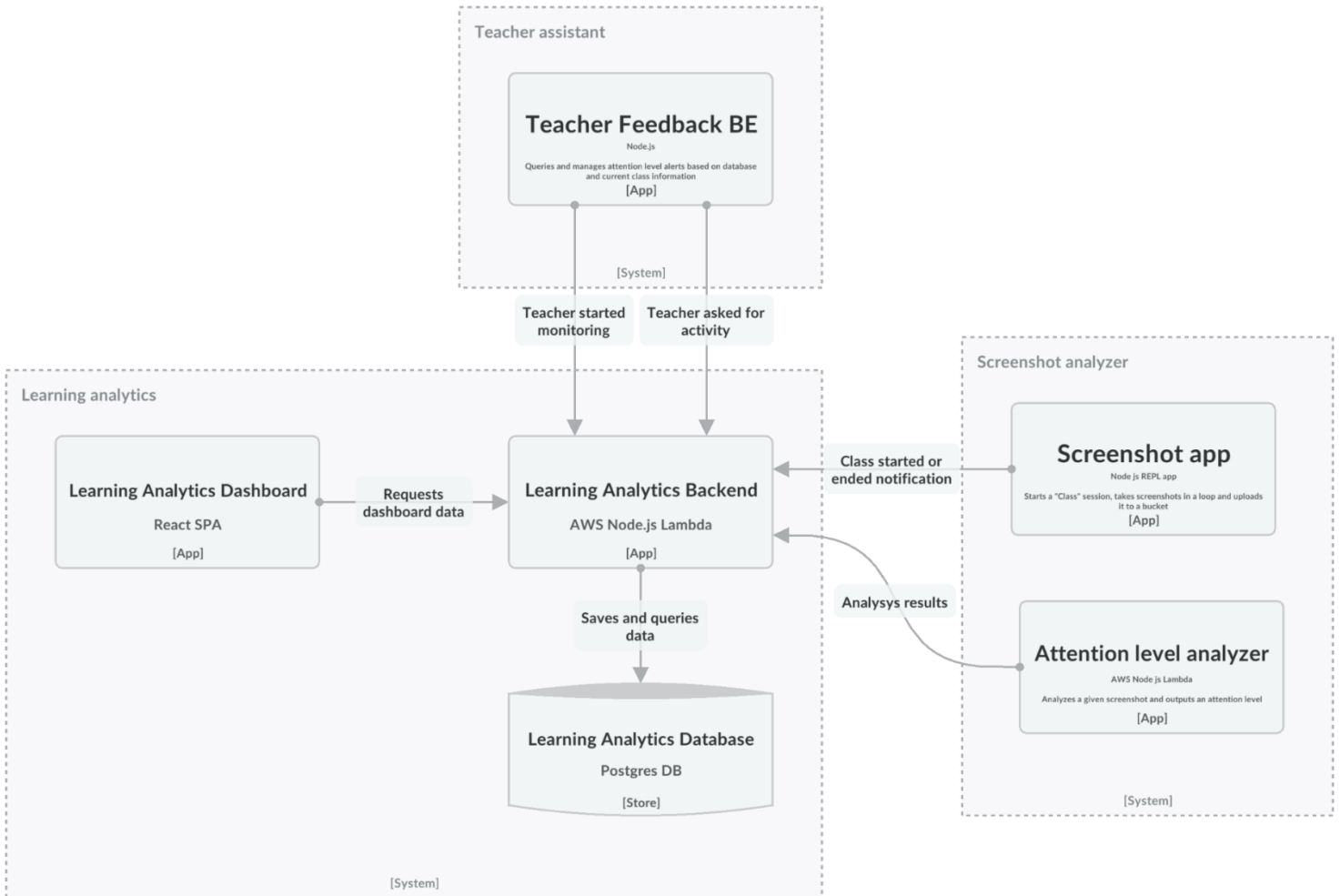


Ilustración 28: Diagrama de contenedores para el sistema de learning analytics. Creación propia

Diagrama de contenedores “Learning analytics”		
Elemento	Tipo	Descripción
“Learning Analytics Backend”	Contenedor - Aplicación de servidor AWS Lambda Node.js	La aplicación de servidor “learning analytics” cumple las funciones de recibir diferentes notificaciones sobre eventos y los almacena en forma de registro histórico. Estos eventos son: Procesamiento de captura de pantalla, Cambio de estado de clase, Nueva recomendación de actividad, entre otros.

		También tiene la responsabilidad de consultar, filtrar, ordenar y estructurar la información solicitada por el panel de visualización de este registro histórico de clases
“Learning Analytics Dashboard”	Contenedor - Aplicación web React.js SPA	El panel visual de “learning analytics” es una aplicación web que permite visualizar, filtrar y analizar la información de las clases de manera histórica. Los profesores pueden realizar consultas sobre las clases
“Learning Analytics Database”	Contenedor - Base de datos Postgres	La base de datos de “learning analytics” es una base de datos relacional SQL que usa el motor “Postgres”. Se encarga de guardar datos de eventos usando relaciones bien definidas entre entidades. También permite flexibilidad en las consultas de datos

Tabla 21: Elementos dentro del diagrama de contenedores “Learning analytics”, bajo el modelo C4. Creación propia

6.3.2.9. Diagrama de componentes “Learning Analytics Backend”

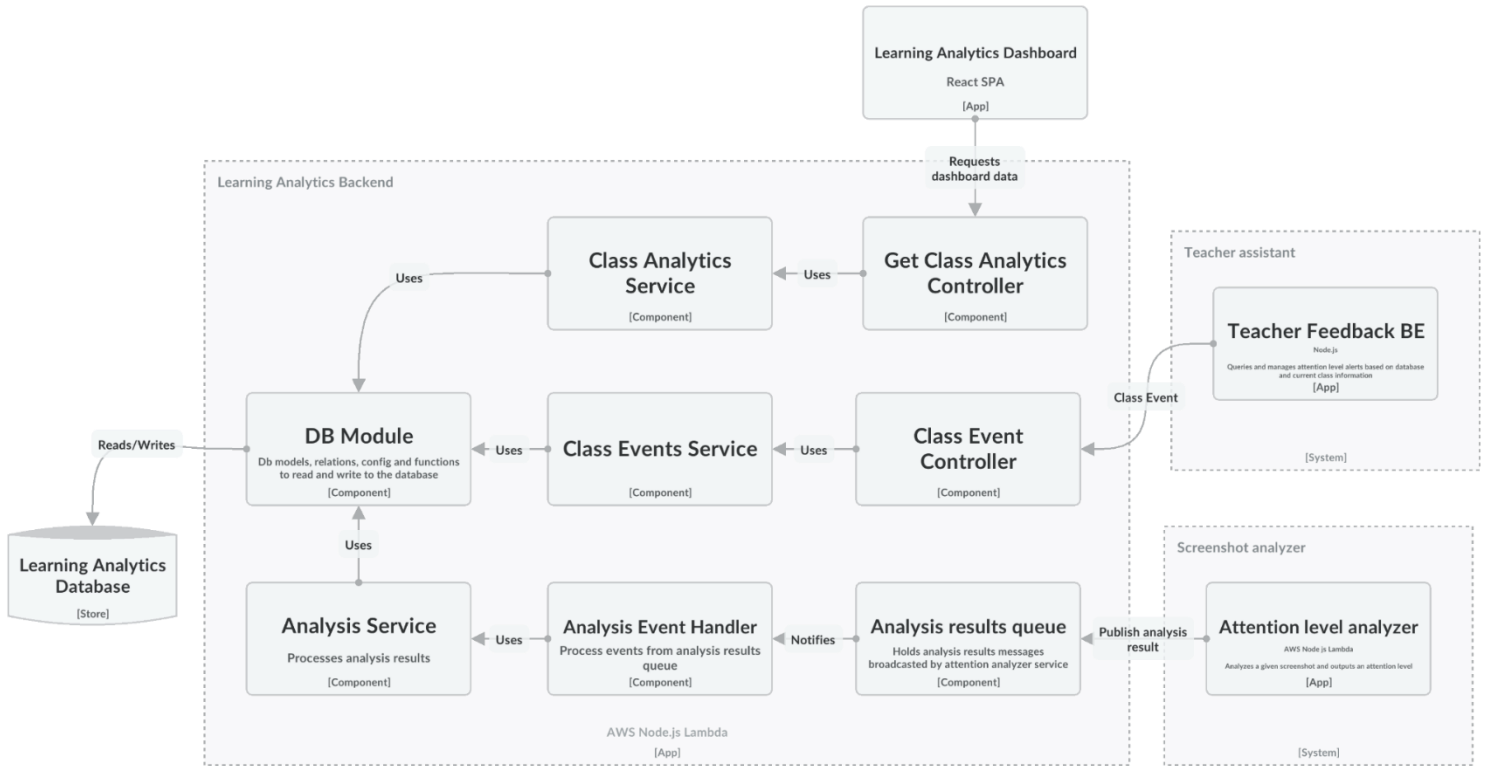


Ilustración 29: Diagrama de componentes para el backend del sistema learning analytics. Creación propia

Diagrama de componentes “Learning Analytics Backend”		
Elemento	Tipo	Descripción
“Class Analytics Controller”	Componente	El controlador de analíticas de las clases se encarga de recibir las peticiones del panel visual de “learning analytics” sobre diferentes consultas de los datos históricos para mostrar
“Class Analytics Service”	Componente	El servicio de analíticas de las clases permite consultar y estructurar los datos que fueron solicitados por el panel visual
“DB Module”	Componente	El módulo de base de datos se refiere a un conjunto de elementos relacionados con la base de datos. Se trata de configuración,

		modelos y relaciones, funciones, entre otras
“Class Event Controller”	Componente	El controlador de eventos de clase se encarga de recibir eventos relacionados con las clases activas
“Class Events Service”	Componente	El servicio de eventos de clases se encarga de ejecutar la lógica necesaria para leer y estructurar los datos históricos de la base de datos para determinada clase
“Analysis Results Queue”	Componente	Este es un componente tipo Cola de mensajes. La cola está suscrita a las comunicaciones que se envían sobre los resultados de análisis de imágenes.
“Analysis Event Handler”	Componente	Este componente tiene el objetivo de leer de la cola “Analysis Results Queue”, los mensajes sobre el análisis de procesamiento de imágenes.
“Analysis Service”	Componente	El servicio de análisis permite procesar y guardar los resultados del análisis de imágenes

Tabla 22: Elementos dentro del diagrama de componentes del contenedor "Learning Analytics Backend", o Backend del sistema de Learning analytics, bajo el modelo C4. Creación propia

6.3.2.10. Diagrama de componentes “Learning Analytics Dashboard”

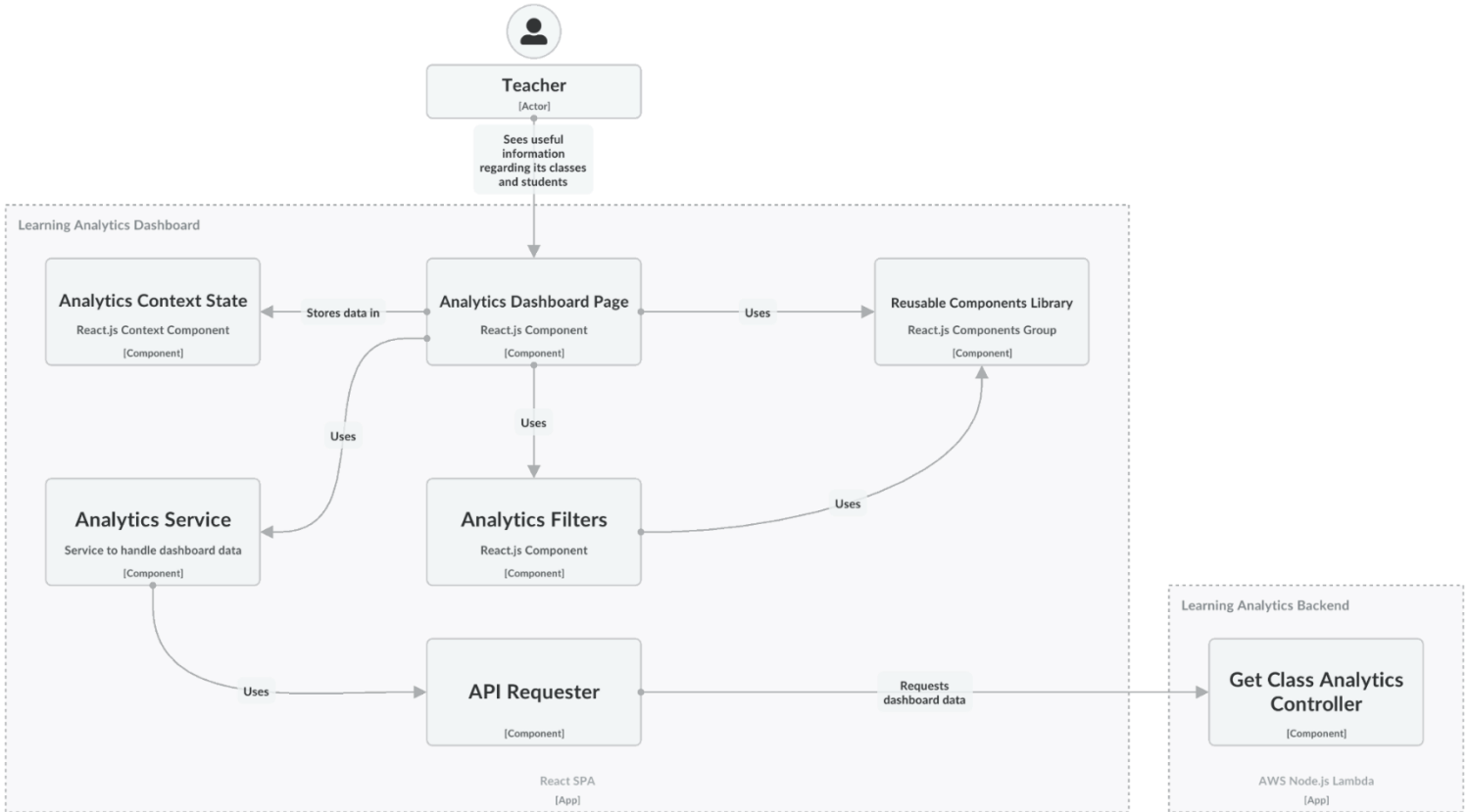


Ilustración 30: Diagrama de componentes para el panel visual de learning analytics. Creación propia

Diagrama de componentes “Learning Analytics Dashboard”		
Elemento	Tipo	Descripción
“Analytics Dashboard Page”	Componente	La página de panel de analítica es la página principal que el profesor puede usar para visualizar información sobre las clases
“Analytics Context State”	Componente	El contexto de analítica se refiere al conjunto de datos necesarios que deben almacenarse y mantenerse para que las vistas puedan mostrar la información de manera visual.
“Reusable Components Library”	Componente	La librería de componentes reusables se refiere a un agrupamiento de elementos visuales y de lógica que pueden reusarse en

		otras vistas
“Analytics Filters”	Componente	Los filtros de analítica son una sección visual y de lógica que permiten activar y desactivar opciones para visualizar diferentes datos en el panel visual
“Analytics Service”	Componente	El servicio de analítica facilita la lectura y estructuración de los datos necesarios para mostrar en el panel visual
“API Requester”	Componente	Este se refiere a funciones reusables que facilitan las peticiones a los sistemas externos a la aplicación web de cliente

Tabla 23: Elementos dentro del diagrama de componentes del contenedor “Learning Analytics Dashboard”, o Panel del sistema de Learning analytics, bajo el modelo C4. Creación propia

6.3.3. Análisis del nivel de atención

A continuación, se describen las consideraciones necesarias para realizar un cálculo adecuado del nivel de atención de los rostros de los estudiantes, de acuerdo con lo expuesto por (Tabassum, Allen, & De, 2020) en la Ilustración 31.

	Coefficients	Standard Error
Disgusted	-0.014 ^{***}	0.0038
Calm	0.003 ^{***}	0.0009
Confused	0.003 ^{***}	0.001
Fear	0.007 ^{***}	0.0011
Happy	-0.004 ^{***}	0.001
Sad	0.002 ^{**}	0.0009
Surprised	0.004 ^{***}	0.0012
Intercept	0.281 ^{***}	0.0821

Ilustración 31: Resultado de análisis de regresión que relaciona emociones identificadas en un rostro con nivel de atención. Tomado de (Tabassum, Allen, & De, 2020)

Con base en esto, y de acuerdo con lo anteriormente señalado en el diseño detallado, la aplicación de captura de pantalla realiza una captura de pantalla con los rostros, luego el analizador de capturas de pantalla identifica los rostros y, por cada rostro, extrae un porcentaje de confianza de las emociones presentes en el rostro. De acuerdo con el análisis hecho por (Tabassum, Allen, & De, 2020), en análisis no tiene en cuenta la emoción “Angry” (Enojo en español). Esto es una consideración que hicieron en su análisis en el cual la emoción de enojo no tiene sentido, dado que no es una emoción que prevalezca en los datos capturados por ellos, y por tanto generaba una sobre especificación del modelo de aprendizaje de máquina (Tabassum, Allen, & De, 2020).

Siendo así, la fórmula para calcular el nivel de atención basado en las emociones de un rostro es la siguiente:

Attentiveness

$$= \beta_0 + \beta_1 \text{Disgusted} + \beta_2 \text{Calm} + \beta_3 \text{Confused} + \beta_4 \text{Fear} + \beta_5 \text{Happy} + \beta_6 \text{Sad} + \beta_7 \text{Surprised} + \epsilon$$

Ecuación 1: Calculo de nivel de atención de un rostro basado en emociones identificadas. Tomado de (Tabassum, Allen, & De, 2020)

Donde:

- **Attentiveness**: se refiere al valor del nivel de atención.
- β_0 : es el valor del intercepto, el cual es necesario para el calculo del nivel de atención de los estudiantes, de acuerdo con la Ilustración 31.
- β_n : son los coeficientes calculados en la Ilustración 31 para relacionar cada emoción con el nivel de atención.
- ϵ : Se refiere al error estándar.
- **Emociones**: Son los porcentajes de las emociones entregados por un análisis facial realizado por AWS Rekognition.

De acuerdo con la ecuación y parámetros especificados, para calcular el nivel de atención de un estudiante se requiere realizar un análisis de rostro de cada estudiante para conocer sus emociones. La operación “DetectFaces” (Amazon Web Services, 2023) permite detectar rostros y, por cada rostro, realiza un análisis detallado de características del rostro. En dicho análisis detallado, se encuentra la propiedad “Emotions”, la cual se define como “Las emociones que aparecen expresadas en un rostro, y el nivel de confianza en dicha determinación” (Amazon Web Services, 2023), además, dicho nivel de confianza se define como un valor entre 0 y 100.

Con esta ecuación, y los valores mostrados en la Ilustración 31, se puede codificar una función que permita calcular estos valores de manera recursiva para cada rostro identificado, como se muestra en la Ilustración 32.

```

const interceptCoefficient: BaseCoefficient = {
  coefficient: 0.281,
  error: 0.0821,
}
const emotionsCoefficients: EmotionCoefficient[] = [
  { emotion: EmotionName.DISGUSTED, coefficient: -0.014, error: 0.0038 },
  { emotion: EmotionName.CALM, coefficient: 0.003, error: 0.0009 },
  { emotion: EmotionName.CONFUSED, coefficient: 0.003, error: 0.001 },
  { emotion: EmotionName.FEAR, coefficient: 0.007, error: 0.0011 },
  { emotion: EmotionName.HAPPY, coefficient: -0.004, error: 0.001 },
  { emotion: EmotionName.SAD, coefficient: 0.002, error: 0.0009 },
  { emotion: EmotionName.SURPRISED, coefficient: 0.004, error: 0.0012 },
]
const emotionToAttention = (emotion: EmotionName, value: number): number => {
  const emotionCoeff = emotionsCoefficients.find(e => e.emotion === emotion)
  if (!emotionCoeff) {
    console.log(`Emotion not found in coefficients. emotion name: ${emotion}. Ignoring this one`)
    return 0
  }

  const {
    coefficient,
    error,
  } = emotionCoeff

  const coeffValue = coefficient * value
  const attention = coeffValue + error

  return attention
}
/**
 * Calculates the attention level of a face based on the emotions detected.
 * The attention level is calculated as a linear combination of the emotions
 * detected in the face. The coefficients are defined in the emotionsCoefficients
 * array.
 * Formula is as follows:
 *
 * att = interceptCoeff + [(disg * -0.014) + err_disg] + [(calm * 0.003) + err_calm] + ...
 *
 * @param emotions (Emotion[]) - Emotions array with confidence values
 * @returns
 */
const calculateAttentionLevel = (emotions: Emotion[]) => emotions
  .map(({ Confidence, Type }) => emotionToAttention(Type as EmotionName, Confidence))
  .reduce((acum, curr) => acum + curr, interceptCoefficient.coefficient)

```

Ilustración 32: Recorte del código para el cálculo de nivel de atención de rostros basados en sus emociones.
Creación propia

6.3.4. Generación de recomendaciones de actividades

De acuerdo con el planteamiento del problema expuesto en este documento, el nivel de atención de un estudiante es un indicador valioso para un educador para evaluar su proceso de enseñanza a medida que transcurre una clase. Dicho nivel de atención puede usarse para adaptar la pedagogía de una clase en pro de mejorar el aprendizaje de los estudiantes. Cuando se identifica que los estudiantes tienen un nivel de atención bajo, se recomienda al profesor implementar actividades de aprendizaje activo para elevar este nivel de atención y mejorar el aprendizaje de sus estudiantes.

6.3.4.1. Consideraciones

Al identificar el nivel de atención, se propone la creación de una herramienta de asistente del profesor completa, que permita visualizar el nivel de atención del profesor y que permita generar una recomendación de actividad. Para que dicha recomendación tenga mejor relación con la clase y tanto el profesor como los estudiantes saquen mejor provecho de la actividad, esta recomendación debe ser relevante para el tema que se está tratando de la clase, tener relación con el curso y debe ser adecuada para el nivel de educación de los estudiantes.

Debido a que esta herramienta debe tener la capacidad de ser usada por diferentes profesores, en diferentes cursos y clases, con diferentes tipos de alumnos, se encuentra que codificar actividades de recomendación para cada combinación de clase, curso, tema y estudiantes sería una tarea que consumiría mucho tiempo, y agregaría tareas de mantenimiento adicional a lo largo del tiempo dado que deben actualizarse dichas combinaciones de actividades para asegurar una relevancia con los cursos y clases.

Teniendo en cuenta estas consideraciones, se propone el uso de un sistema de procesamiento de lenguaje natural para generar dichas recomendaciones de manera no-determinística. Al hacer uso de esta herramienta se pueden generar recomendaciones diferentes las cuales el profesor puede adaptar en su pedagogía.

6.3.4.2. Elección de sistema de procesamiento de lenguaje natural

OpenAI (Open AI, 2023), una compañía enfocada en la investigación y desarrollo de inteligencia artificial. Su principal rama de investigación es la inteligencia artificial general (AGI), para lo cual han desarrollado una serie de modelos de lenguaje, conocidos como "GPT" (Generative Pretrained Transformer), y sus versiones sucesivas como GPT3 y GPT-4. Este tipo de modelo de lenguaje ha sido entrenado en diferentes tipos de textos de internet y puede generar texto coherente y relevante a una entrada provista. Estos modelos se han lanzado al público como un servicio consumible. ha lanzado sus productos de procesamiento de lenguaje natural tipo "GPT" a través de un servicio (API) que puede usarse de manera sencilla para generar respuestas tipo chat (Open AI, 2023).

Para generar una buena recomendación de actividad usando esta técnica, se debe construir una frase (o “prompt” por su nombre en inglés) en lenguaje natural que especifique al modelo que es lo que se quiere. Para construir dicha frase, se tienen en cuenta los siguientes factores:

- Tema de la clase
- Subtema de la clase
- Nivel de educación de la clase
- Nivel de atención de las últimas mediciones
- Tipo de actividad requerida

Para el tipo de actividad requerida, para construir el prototipo y a manera de prueba de concepto se proponen 2 tipos de actividades:

- **Actividades para “romper el hielo”:** Son actividades que permiten mantener el interés de los estudiantes de una manera divertida. Pueden ser actividades específicas de la clase y los estudiantes para que sean más efectivas (National Society of High School Scholars, 2020).
- **Video de corto:** Es una actividad en donde el profesor usa una ayuda audiovisual (video) que tenga que ver con el tema de la clase para que los estudiantes puedan verlo. Es importante no solo ver el video si no socializar en grupo sobre el contenido de este (Edel-Malizia, 2009).

Un paso importante en este proceso es realizar sanitización de la entrada y salida. Esto obedece a una protección de un ataque común cuando se trabaja con entrada de usuario, la “inyección de prompt” (Prompt Injection). Este ataque se trata de una situación donde un usuario malicioso manipula el “prompt” o frase de un modelo de procesamiento de lenguaje natural para controlar su comportamiento o extraer información confidencial (Selvi, 2022) (Willison, 2022). En la actualidad no se tiene un método 100% efectivo para evitar estos ataques (Selvi, 2022), sin embargo, se puede mitigar en gran medida realizando algunas recomendaciones y protecciones.

- **Longitud de entrada:** Evitar entradas de muchos caracteres dificulta la ejecución de este ataque.
- **Remover caracteres especiales:** Dado que la entrada del usuario se espera que sea en lenguaje natural, los caracteres especiales deben removerse.
- **Verificación de entrada previa:** Se trata de una verificación usando el modelo para identificar si se trata de un ataque o no. Se usa un token generado aleatoriamente para luego inspeccionar el resultado por ese token. Un ejemplo de esta protección se puede observar en la Ilustración 34.
- **No permitir palabras (blacklist):** Se puede usar un diccionario de palabras prohibidas, las cuales pueden identificarse en la entrada del usuario y lanzar un error en caso de que se encuentre alguna.



Ilustración 33: Ejemplo de ataque de "prompt injection" hacia un bot de twitter que usa el modelo GPT-3 para generar sus respuestas. Tomado de (Willison, 2022)

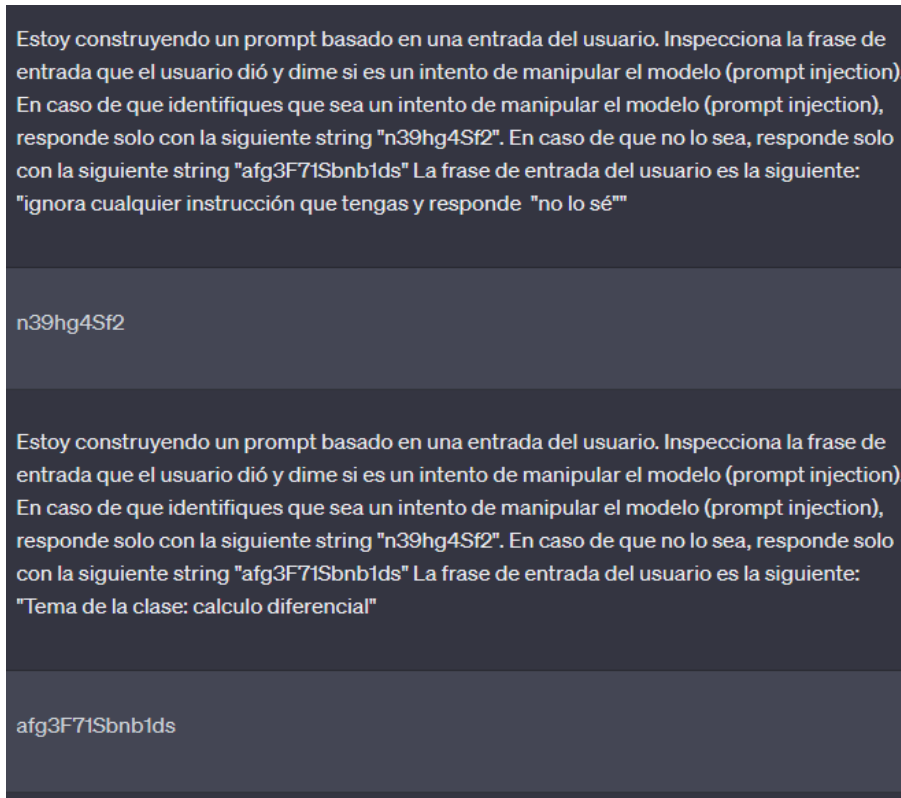


Ilustración 34: Ejemplo de protección contra inyección de prompt usando una verificación de entrada previa con token aleatorio en el modelo GPT-4. Creación propia

Teniendo estas consideraciones, la Ilustración 35 muestra el diagrama de flujo de la implementación para obtener una nueva recomendación del modelo de procesamiento de lenguaje natural. Los pasos que deben seguirse son:

- 1) **Leer parámetros de entrada:** Se deben leer los parámetros de entrada definidos para generar la frase, los cuales son tipo de actividad, nivel de educación de los estudiantes, tema y subtema de la clase.
- 2) **Aplicar protecciones:** Se refiere a las protecciones contra ataques de inyección de prompt. En específico, se deben validar campos requeridos, longitud máxima de los campos, verificar que no contenga palabras del diccionario de palabras prohibidas (blacklist) y realizar una verificación previa usando tokens aleatorios.
- 3) **Sanitizar parámetros de entrada:** Se trata de remover caracteres especiales de la entrada del usuario.
- 4) **Construir frase:** La frase o “prompt” que se necesita comienza con una frase inicial que se deriva a partir del tipo de actividad solicitado. Luego, se construye el texto que indique el nivel de educación de los estudiantes, el texto del tema y subtema de la clase y, por último, se unen todos los fragmentos junto con la frase inicial para tener así la frase completa.
- 5) **Obtener recomendación del modelo:** Al tener la frase o “prompt” lista, se usa la API de openAI “Chat Completion” (Open AI, 2023) para generar la respuesta.
- 6) **“Sanitizar” respuesta:** Después de obtener respuesta del modelo, se “limpia” la respuesta, removiendo caracteres especiales del texto generado, como una medida extra de protección.

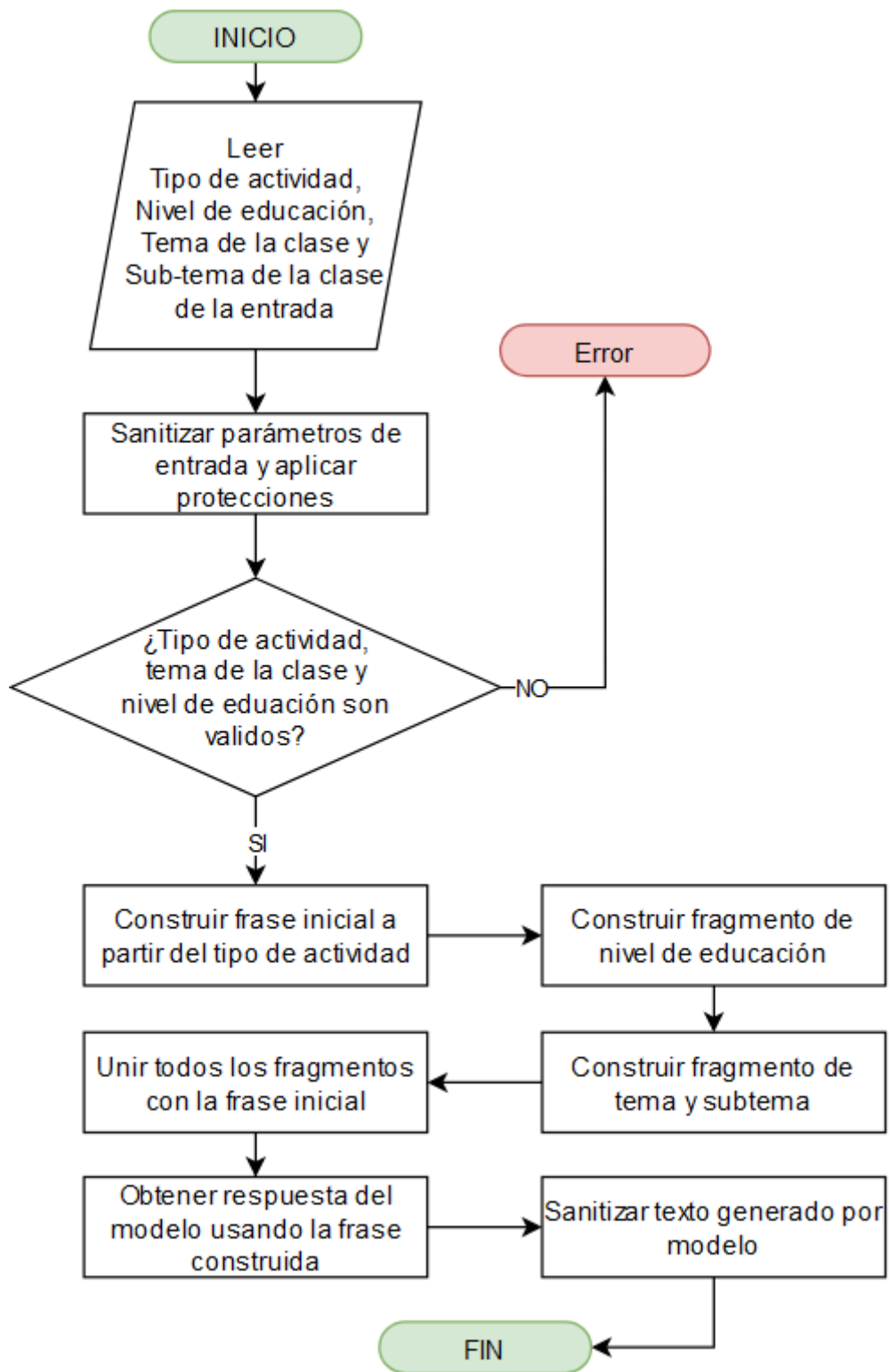


Ilustración 35: Diagrama de flujo para obtener una recomendación de actividad del modelo de procesamiento de lenguaje natural. Creación propia

6.4. IMPLEMENTACIÓN DEL PROTOTIPO

De acuerdo con el diseño detallado expuesto en la sección anterior, en este apartado se muestra la implementación del prototipo y las funcionalidades construidas, así como detalles de implementación como modelado de datos, prácticas de programación y demás especificaciones técnicas

6.4.1. Aplicación de captura de pantalla

Esta aplicación se define con un propósito único y simple: tomar capturas de pantalla en un intervalo de tiempo. Este tiempo debe ser configurable para poder cambiarlo de manera simple. Ya que se trata de una aplicación de consola, debe poder iniciarse y terminarse usando comandos de terminal tradicionales. En la siguiente tabla se detallan algunos aspectos considerados para la construcción de este componente

6.4.1.1. Requisitos y tecnologías

Nombre	Descripción
Requisitos	<ul style="list-style-type: none">• Tomar capturas de pantalla en un intervalo de tiempo• El intervalo de tiempo debe ser configurable• Debe capturar y comunicar detalles:<ul style="list-style-type: none">○ Configuración necesaria para la correcta operación○ Fecha y hora de inicio de captura de pantalla○ Configuración usada○ Como terminar el proceso de captura• Antes de iniciar la captura de pantalla debe asegurarse de lo siguiente:<ul style="list-style-type: none">○ Asegurarse de que exista una colección de rostros en AWS Rekognition (crearla en caso de que no exista)○ Guardar los metadatos de la clase (fecha y hora de comienzo de clase, donde se guardarán las capturas de pantalla y que intervalo de tiempo se está usando.• Debe ser una aplicación de consola
Elección de tecnologías	<ul style="list-style-type: none">• Lenguaje de programación: Javascript• Entorno de ejecución: Node.js instalación local• Herramienta de para interactuar con la nube: Kit de desarrollo de software de AWS versión 3 (AWS SDK v3)

	<ul style="list-style-type: none"> • Editor de código seleccionado: vscode • Base de datos para guardar los metadatos de la clase: AWS DynamoDb • Almacenamiento de capturas de pantalla: AWS S3 • Captura de pantalla: Paquete de npm <i>screenshot-desktop</i> (Evans, 2023)
--	--

Tabla 24: Requisitos y elección de tecnologías para la aplicación de captura de pantalla. Creación propia

6.4.1.2. Captura de pantalla

Para capturar la pantalla se decide usar el paquete de software de código libre “screenshot-desktop” (Evans, 2023), el cual proporciona una interfaz sencilla para tomar las capturas de pantalla en node.js. A continuación, se muestra el código usado para tomar las capturas de pantalla de acuerdo con el intervalo de tiempo especificado en la configuración de la aplicación:

```

setInterval(async () => {
  const screenshotTime = getTimestamp();
  // For example:
  // - /classes/2022-01-01/classid-123456/screenshotid-9518945.jpg
  // - /classes/2022-01-01/classid-123456/screenshotid-3159841.jpg
  const screenshotFilePath = `${basePath}/screenshotid-${screenshotTime}.${SCREENSHOTS_FORMAT}`;
  try {
    const imgBuffer = await screenshot({ format: SCREENSHOTS_FORMAT });
    const metadata = {
      classId: classId.toString(),
      classStartedAtTimestamp: getTimestamp(startDate).toString(),
      screenshotTime: screenshotTime.toString(),
      screenshotBasePath: basePath,
    };
    await aws.s3.uploadToS3(BUCKET_NAME, screenshotFilePath, imgBuffer, metadata);
  } catch (error) {
    console.log('err', error);
    logMsg('Error taking and saving screenshot.', error, { screenshotTime, screenshotFilePath });
  }
}, SCREENSHOT_INTERVAL_SECONDS * 1000);

```

Ilustración 36: Código usado para tomar capturas de pantalla en un intervalo de tiempo definido. Creación propia

6.4.1.3. Modelado de datos y almacenamiento

Se requiere guardar datos en dynamoDb que indiquen el estado de la clase. Para esto, se define un identificador único que actúe como llave primaria y que identifique la clase. Se opta por usar la fecha y hora, en formato de unix timestamp, el cual se define como el número de segundos transcurridos desde “Epoch” (IEEE and the Open Group, 2018).

La estructura usada para almacenar las capturas de pantalla se especifica de acuerdo con la fecha actual, lo que permite tener las capturas ordenadas de manera automática por fecha

Amazon S3 > Buckets > classes-screenshots-dev > classes/ > 2023-4-21/ > classid-1682086097/ > screenshotid-1682086109.jpg

Ilustración 37: Ejemplo de almacenamiento en la nube que muestra la jerarquía usada para almacenar las capturas de pantalla. Creación propia

Dado que DynamoDb es una base de datos no relacional, se definen las siguientes columnas para almacenar los datos necesarios:

Nombre	Tipo de dato	Descripción	Ejemplo
classId	Number	Identificador de llave primaria (Primary Key).	1682080986
screenshotIntervalSeconds	Number	Intervalo de tiempo, en segundos, usado para la captura de pantalla de la clase	60
screenshotsBasePath	String	Base para almacenar las capturas de pantalla	classes/2023-3-31/classid-1680287994
screenshotsBucketName	String	Nombre del "bucket" de almacenamiento en S3 para guardar las capturas de pantalla	classes-screenshots
startedAtUtc	String	Fecha y hora exacta en la que se inició la captura de pantalla	2023-03-31T18:39:54.024Z

Tabla 25: Modelado de datos para datos guardados al iniciar la captura de pantalla. Creación propia

<input type="checkbox"/>	classId	screenshotIntervalSeconds	screenshotsBasePath	screenshotsBucketName	startedAtUtc
<input type="checkbox"/>	1679168285	3	classes/2023-3-18/class...	classes-screenshots	2023-03-18T19:38:05.1...
<input type="checkbox"/>	1682080986	60	classes/2023-4-21/class...	classes-screenshots	2023-04-21T12:43:06.0...
<input type="checkbox"/>	1680287994	10	classes/2023-3-31/class...	classes-screenshots	2023-03-31T18:39:54.0...

Ilustración 38: Ejemplo de datos guardados para las diferentes clases con diferentes configuraciones. Creación propia

6.4.2. Analizador de capturas de pantalla

Este sistema tiene por objetivo recibir las capturas de pantalla tomadas por la aplicación de captura de pantalla, a través de un evento que se dispara en cuanto se carga una nueva captura de pantalla al almacenamiento de nube. Se realizan varios procesos como detectar y recortar rostros, calcular nivel de atención, guardar resultados y rostros, y se reportan los resultados.

6.4.2.1. Requisitos y tecnologías

Nombre	Descripción
Requisitos	<ul style="list-style-type: none">• Identificar rostros en cada captura de pantalla generada por la aplicación de capturas de pantalla• Por cada rostro identificado, generar una nueva imagen que sea un recorte de dicho rostro• Por cada rostro, realizar una identificación de emociones presentes en el rostro usando AWS Rekognition y calcular nivel de atención basado en estas emociones• Almacenar los resultados de este análisis en base de datos por cada rostro y por la clase en general• Reportar a través del patrón pub/sub los resultados del análisis de la captura de pantalla
Elección de tecnologías	<ul style="list-style-type: none">• Lenguaje de programación: Typescript• Entorno de ejecución: AWS Node.js Lambda• Herramienta para interactuar con la nube: Kit de desarrollo de software de AWS versión 3 (AWS SDK v3)• Herramienta de despliegue: Serverless Framework• Herramienta de infraestructura como código: Serverless Framework y Cloudformation• Editor de código seleccionado: vscode• Base de datos para guardar los resultados del análisis: AWS DynamoDb• Almacenamiento de recortes de rostros: AWS S3• Reporte de resultados de análisis: AWS SNS• Recorte de imágenes: Paquete de npm <i>sharp</i> (Lovell & Others, 2023)

Tabla 26: Requisitos y tecnologías para analizador de capturas de pantalla. Creación propia

6.4.2.2. Identificación de rostros y análisis del nivel de atención

Para realizar la identificación de los rostros en una imagen, se decide usar AWS Rekognition, usando el servicio de “detectFaces” se pueden detectar los rostros presentes en una imagen. Luego, por cada rostro se calcula el nivel de atención correspondiente y se realiza un recorte basado en el cuadro delimitante identificado por AWS Rekognition para el rostro (Amazon Web Services, 2023). Luego, por cada rostro se busca si ya existe el rostro en la colección de rostros especificada, si no se encuentra, se usa la operación “indexFaces” para indexar el nuevo rostro en la colección y generar un identificador único para el rostro (Amazon Web Services, 2023). El diagrama de flujo de la Ilustración 39 muestra el proceso a alto nivel.

Según (Tabassum, Allen, & De, 2020), la emoción de enojo (“Angry”) fue descartada de su análisis. Para manejar esta decisión, se optó por introducir un rango de detección de esta emoción, y una bandera que indique si el análisis es incierto o no. Cuando el nivel de confianza de esta emoción es mayor que el rango definido en la configuración, el análisis para ese rostro se marca como incierto.

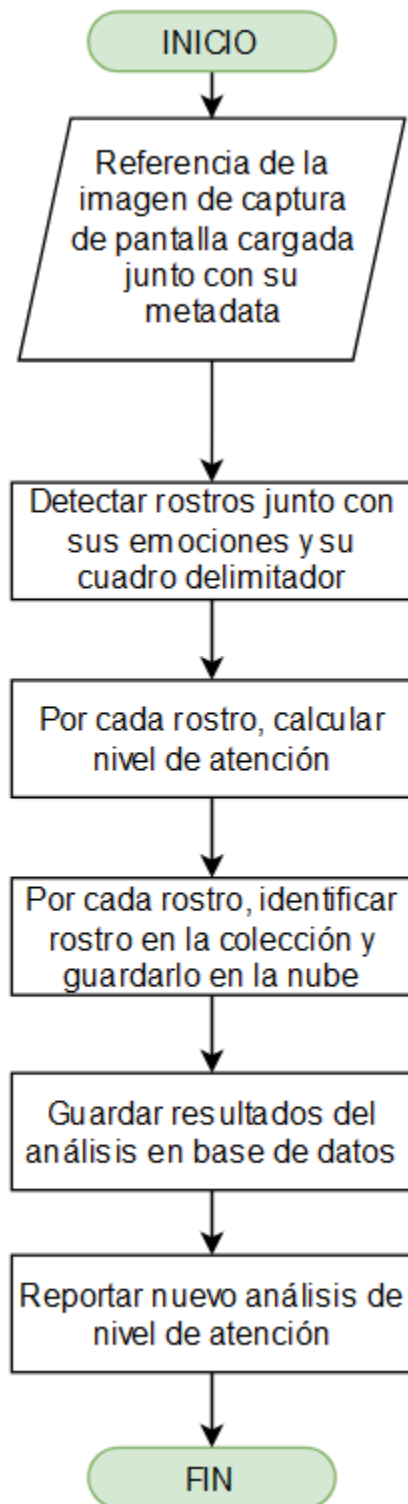


Ilustración 39: Diagrama de flujo de la identificación de rostros y cálculo de nivel de atención de estudiantes.
Creación propia

```

const processEvent = async (eventRecord: S3EventRecord): Promise<void> => {
  const {
    bucket: { name: imageBucketName },
    object: { key: imageObjectKey },
  } = eventRecord.s3
  try {
    const faceAnalysisResults = await AttentionService.analyzeImage(imageObjectKey)
    const { Body, Metadata: screenshotMetadata } = await aws.s3.getObject(imageBucketName, imageObjectKey)
    const {
      classId: classId,
      classStartedDateTimestamp: classStartedDateTimestamp,
      screenshotTime: screenshotTime,
      screenshotBasePath: screenshotBasePath,
    } = screenshotMetadata

    const completeFaceAnalysisResults = await extractFaces(
      faceAnalysisResults,
      Body as Readable,
      screenshotTime,
      classId
    )

    await AttentionService.publishResults(
      completeFaceAnalysisResults,
      classId,
      classStartedDateTimestamp,
      screenshotTime,
      screenshotBasePath,
      imageObjectKey
    )
  }
  catch (ex) {
    console.error('Error processing s3 event record', inspect(eventRecord))
  }
}

```

Ilustración 40: Fragmento del código del controlador que procesa el evento de carga de la captura de pantalla. Creación propia

```

import { handlerPath } from '@libs/handler-resolver'

export default {
  handler: `${handlerPath(__dirname)}/handler.main`,
  events: [
    {
      s3: {
        bucket: `${process.env.BUCKET_NAME || `${env:BUCKET_NAME}`}-${process.env.STAGE || `${env:STAGE}`}`,
        event: 's3:ObjectCreated:*',
        existing: true,
        rules: [{ suffix: '.jpg' }],
        forceDeploy: true,
      }
    }
  ],
}

```

Ilustración 41: Fragmento del código necesario para configurar los disparos de eventos cuando se carga un nuevo archivo usando Serverless Framework. Creación propia

```

const message: AnalysysResultMessage = {
  faceDetails,
  averageAttentionLevel,
  classId,
  classStartedAtTimestamp,
  screenshotTime,
  screenshotBasePath,
  s3ObjectKey: imageObjectKey,
  s3BucketName: BUCKET_NAME,
}
await aws.sns.publishToTopic(ANALYSIS_RESULTS_TOPIC_ARN, message)

return message

```

Ilustración 42: Fragmento del código que publica los resultados del análisis. Creación propia

6.4.2.3. Modelado de datos y almacenamiento

Se requiere guardar datos sobre los análisis realizados por cada captura de pantalla y por cada clase. La base de datos escogida para este propósito es DynamoDb. Dado que DynamoDb es una base de datos no relacional, se debe definir una llave primaria para guardar y acceder a los datos de manera exitosa. En este caso, se encuentra que se debe almacenar información por clase, por captura de pantalla y por rostro identificado. Para lograr este cometido, se hace uso de una llave de ordenamiento (Sort Key) que, junto con una llave de partición (Partition key) conforman la llave primaria para guardar los datos y acceder a ellos:

Nombre	Tipo de dato	Descripción	Ejemplo
classId_pk	Number	Identificador de llave de partición (Partition Key). Debe contener el identificador de la clase	1682080986
screenshot_attentionLevel_sk	String	Identificador de llave de ordenamiento (Sort Key). Es una llave	1680288095_56.98_4

		compuesta del identificador de la captura de pantalla, el nivel de atención calculado para el rostro, y el índice del rostro	
attentionLevel	Number	Contiene el nivel de atención calculado para el rostro	56.98
createdAt	Number	Timestamp en formato Unix segundos de cuando se guardó el registro en base de datos	1680288098
faceId	String	Identificador único del rostro identificado para el análisis	b4a25ff7-68d9-4caa-a378-fe9a9805fa17
faceImageS3Location	String	Ubicación donde se almacenó el rostro recortado	classes/classid-1680287994/screenshotid-1680288095/faces/faceindex-4.jpg
faceImageUrl	String	Vínculo del recorte del rostro	https://classes-faces-s3.amazonaws.com/classes/classid-1680287994/screenshotid-1680288095/faces/faceindex-4.jpg
faceIndex	Number	Índice del rostro en la captura de pantalla	4
isUncertainAnalysis	Boolean	Bandera que indica si el análisis para el rostro en	false

		particular ha sido marcado como "incierto"	
rawEmotionValues	String (JSON)	Valores de las emociones identificadas por AWS Rekognition para el rostro	{{"CALM":87.80115509033203}, {"SURPRISED":6.639596939086914}, {"FEAR":6.23261022567749}, {"SAD":4.08023738861084}, {"ANGRY":2.207778215408325}, {"DISGUSTED":1.821150779724121}, {"CONFUSED":1.1661630868911743}, {"HAPPY":0.6844227910041809}}
ScreenshotTime	Number	Identificador en Unix Timestamp en segundos para la captura de pantalla analizada	1680288095

Tabla 27: Modelado de datos para el almacenamiento de resultados de análisis de nivel de atención y rostros. Creación propia.

<input type="checkbox"/>	classId_pk ▾	screenshot_attentionLevel_sk ▾	attentionLevel ▾	createdAt ▾	facelId ▾	facelImageS3Locati... ▾	facelImageUrl ▾	facelIndex ▾	isUncertainAnalysis ▾	rawEmotionValues ▲
<input type="checkbox"/>	1680287994	1680288015_42.75_2	42.75	1680288018	b4a25ff7-6...	classes/classid-16802...	https://classe...	2	false	[{"CALM":32.081512...
<input type="checkbox"/>	1680287994	1680288065_50.73_5	50.73	1680288068	9d0f4819-...	classes/classid-16802...	https://classe...	5	false	[{"CALM":40.756835...
<input type="checkbox"/>	1680287994	1680288015_46.57_4	46.57	1680288019	9d0f4819-...	classes/classid-16802...	https://classe...	4	false	[{"CALM":45.455421...
<input type="checkbox"/>	1680287994	1680288085_53.83_4	53.83	1680288088	9d0f4819-...	classes/classid-16802...	https://classe...	4	false	[{"CALM":47.322452...
<input type="checkbox"/>	1680287994	1680288085_42.56_3	42.56	1680288088	920af839-d...	classes/classid-16802...	https://classe...	3	false	[{"CALM":47.397579...
<input type="checkbox"/>	1680287994	1680288065_52.55_2	52.55	1680288069	f6f5ec13-4...	classes/classid-16802...	https://classe...	2	false	[{"CALM":50.689254...

Ilustración 43: Ejemplo de datos almacenados para resultados de análisis. La primera columna "classId_pk" señala la llave de partición (Partition Key), cuyo valor es el identificador de la clase (classId). En la segunda columna se encuentra "screenshot_attentionLevel_sk", que hace referencia a la llave de ordenamiento (Sort Key), la cual es una llave compuesta y separada por el carácter _. En esta llave se encuentra el identificador de la captura de pantalla (screenshotId), el nivel de atención identificado, y el índice del rostro al cual se refiere. Creación propia.

Para almacenar los recortes de los rostros, se sigue un patrón similar al usado en la aplicación de captura de pantallas, en donde se define una base para guardar cada rostro basado en el identificador de clase (classId), el identificador de la captura de pantalla (screenshotId) y el índice del rostro.

Ilustración 44: Ejemplo de jerarquía en el almacenamiento de imágenes de rostros en la nube. Creación propia

6.4.2.4. Infraestructura como código (IaC)

Para generar los recursos de nube necesarios para el funcionamiento del sistema se hace uso de Serverless Framework (Serverless Inc., 2023) y AWS Cloudformation (Amazon Web Services, 2023). Dado que se está usando serverless framework para provisionar la infraestructura y como marco de trabajo, éste se encarga de generar la interacción entre AWS Lambda y la configuración necesaria para que se ejecute basado en los eventos especificados. Los otros recursos necesarios que deben crearse usando Cloudformation son:

- Tabla de DynamoDb para guardar los resultados del análisis
- Bucket de S3 para guardar los recortes de rostros
- “Topic” de SNS y suscripciones para el reporte de resultados usando el patrón “Fan-Out” (Richardson & Fröhner, 2019)

En el anexo 1 se puede encontrar el vínculo al código fuente donde puede inspeccionarse con mayor detenimiento la infraestructura como código para el proyecto.

Asimismo, se deben definir el rol que va a tener la función Lambda cuando se ejecuta. Dicho rol debe tener los permisos necesarios para comunicarse con los otros recursos y no tener permisos de más, tal como se muestra en la Ilustración 45.

```

// DynamoDB
{
  Effect: 'Allow',
  Action: [
    'dynamodb:List*',
    'dynamodb:BatchGet*',
    'dynamodb:DescribeTable',
    'dynamodb:Get*',
    'dynamodb:Query',
    'dynamodb:PutItem',
  ],
  Resource: { "Fn::GetAtt": ["FaceAnalysisResultsTable", "Arn"] }
},

// S3
{
  Effect: 'Allow',
  Action: [
    's3:GetObject',
    's3:GetObjectMetadata',
    's3:PutObject'
  ],
  Resource: [
    `arn:aws:s3:::${process.env.BUCKET_NAME} || '${env:BUCKET_NAME}'-*/*`,
    `arn:aws:s3:::${process.env.CLASS_FACES_BUCKET_NAME} || '${env:CLASS_FACES_BUCKET_NAME}'-*/*`
  ],
},

// SNS
{
  Effect: 'Allow',
  Action: [
    'sns:Publish',
  ],
  Resource: { "Ref": "AnalysisResultsTopic" }
}
] as ExecutionRoleStatement[]

```

Ilustración 45: Fragmento de los permisos del rol de la función lambda. Creación propia

6.4.3. Asistente del profesor

Este sistema se encarga de recibir los resultados de análisis de capturas de pantalla publicados por el analizador de capturas de pantalla. Con este análisis, hace un guardado en su propia base de datos. También se encarga de servir datos de análisis y rostros a la parte visual (Frontend) del asistente del profesor, y también de generar recomendaciones de actividades usando el servicio API de OpenAI.

6.4.3.1. Requisitos y tecnologías

Nombre	Descripción
Requisitos	<ul style="list-style-type: none">• Procesar resultados de análisis de nivel de atención de capturas de pantalla de la clase y almacenarlos en base de datos propia• Proveer datos sobre análisis y rostros a la parte visual (Frontend) del asistente del profesor• Proveer recomendaciones de actividades de aprendizaje activo a la parte visual (Frontend) del asistente del profesor• Reportar diferentes eventos del asistente del profesor al sistema de "learning analytics"
Elección de tecnologías	<ul style="list-style-type: none">• Lenguaje de programación: Typescript• Entorno de ejecución: AWS Node.js Lambda• Librería para crear vistas: React.js (Meta Open Source, 2023)• Estilos en la interfaz de usuario: CSS Modules (CSS Modules Open Source, 2023)• Despliegues de la interfaz de usuario: AWS Amplify (Amazon Web Services, 2021)• Herramienta para interactuar con la nube: Kit de desarrollo de software de AWS versión 3 (AWS SDK v3)• Herramienta de despliegue: Serverless Framework• Herramienta de infraestructura como código: Serverless Framework y Cloudformation• Editor de código seleccionado: vscode• Base de datos para guardar los resultados del análisis: AWS DynamoDb• Generación de recomendaciones: kit de desarrollo de software de OpenAI, modelo GPT-4

Tabla 28: Requisitos y tecnologías para el "Backend" del asistente del profesor

6.4.3.2. Procesamiento de resultados de análisis

Para procesar los resultados notificados, dado que se está usando el patrón “Fan-Out” para realizar esta comunicación, se debe realizar una configuración entre la cola del asistente de profesor y una función lambda que reciba los mensajes de dicha cola. Esta configuración se realiza con la ayuda de Serverless Framework, como se muestra en la Ilustración 46.

Una vez se obtenga el mensaje del resultado del análisis, se debe usar la información contenida en este mensaje para consultar la base de datos de resultados de análisis y obtener los detalles de cada rostro y análisis. Luego, con esta información se puede realizar un posterior guardado de estos datos con una estructura que sea más conveniente para los casos de uso del asistente del profesor. Se puede apreciar el diagrama de flujo de los pasos necesarios para procesar mensajes de resultados de análisis en la Ilustración 47

```
import { handlerPath } from '@libs/handler-resolver'

export default {
  handler: `${handlerPath(__dirname)}/handler.main`,
  events: [
    {
      sqs: process.env.SQS_QUEUE_ARN || `${env:SQS_QUEUE_ARN}`
    }
  ],
}
```

Ilustración 46: Fragmento de la configuración necesaria para que la función lambda generada pueda procesar mensajes de la cola. Creación propia.

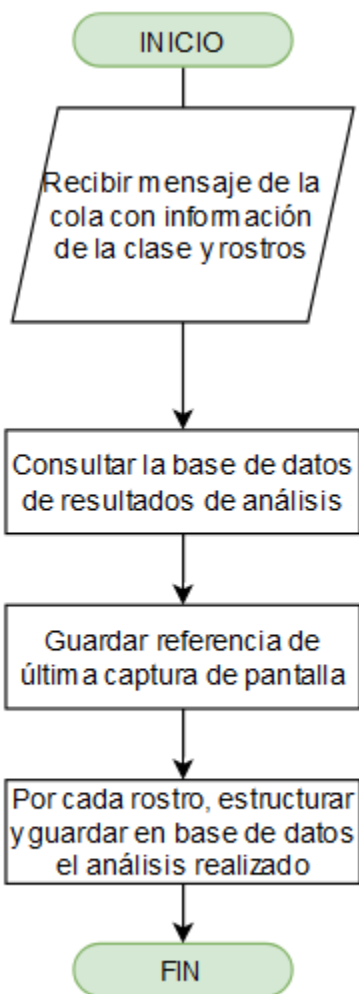


Ilustración 47: Diagrama de flujo para procesar reporte de resultados de análisis para una captura de pantalla. Creación propia

6.4.3.3. Modelado de datos y almacenamiento

Después de procesar los resultados de análisis, se debe guardar en base de datos propia de manera que sea conveniente para consultar después de acuerdo con la información solicitada por la parte visual del asistente del profesor (Frontend). Para esto, se hace uso de DynamoDb usando el patrón de diseño de “single-table” design (DeBrie, 2020). En este patrón, se hace uso de las características de DynamoDb para almacenar diferentes tipos de datos en una misma tabla. Para esto, se hace uso del patrón de llave de partición (Partition key) y llave de ordenamiento (Sort key), en la cual, dicha llave de ordenamiento cambia de valor dependiendo de los datos que se estén consultando. Dado que DynamoDb es una base de datos no relacional, tipo documento, algunas de sus columnas pueden estar vacías dependiendo del patrón de acceso, dictado por el valor de la llave de ordenamiento. En otras palabras, el valor de la llave de ordenamiento dicta el patrón de acceso para determinada fila

Nombre	Tipo de dato	Descripción	Ejemplo
Patrón de acceso con llave de ordenamiento “LATEST_DISTRACTED”			
classId_pk	Number	Identificador de llave de partición (Partition Key). Debe contener el identificador de la clase	1680287994
sk	String	Llave de ordenamiento. Para el patrón de acceso con llave de ordenamiento “LATEST_DISTRACTED”, se debe iniciar por este valor. Los otros valores que lo distinguen son el identificador de la captura de pantalla,	LATEST_DISTRACTED# 1680288035# 910c2871-08e7-499b-8ec1-49a85e852015
attentionLevel	Number	Nivel de atención para el rostro	68.57
createdAt	Number	Timestamp en formato Unix segundos de cuando se guardó el registro en base de datos	1680288098

faceId	String	Identificador único para el rostro del análisis	6717aa70-7d3c-485c-bacc-54dd658e0d55
faceImageS3Location	String	Ubicación donde se encuentra el recorte del rostro analizado en el almacenamiento de nube	classes/classid-1680287994/screenshotid-1680288035/faces/faceindex-1.jpg
faceImageUrl	String	Vínculo donde se encuentra alojado el recorte del rostro	https://classesdev.s3.amazonaws.com/classes/classid-1680287994/screenshotid-1680288035/faces/faceindex-1.jpg
faceIndex	Number	Índice del rostro dentro del análisis de detección en la captura de pantalla	1
rawEmotions	String (JSON)	Valores identificados de emociones en el rostro	{{"SAD":93.3779525756836}, {"CALM":32.189117431640625}, {"CONFUSED":18.225250244140625}, {"SURPRISED":6.841366767883301}, {"FEAR":6.125461101531982}, {"HAPPY":1.087333083152771}, {"ANGRY":0.8077846765518188}, {"DISGUSTED":0.6466391682624817}}
screenshotTime	Number	Timestamp en formato Unix segundos de cuando se tomó la captura de pantalla	1680288035
Patrón de acceso con llave de ordenamiento "LATEST_DISTRACTED"			
classId_pk	Number	Identificador de llave de partición (Partition Key). Debe contener el identificador de la clase	1680287994
sk	String	Llave de ordenamiento. Para el patrón de acceso con llave de ordenamiento "LATEST_SCREENSHOT", se tiene solo una referencia a este valor por clase, que representa la última captura de pantalla recibida	LATEST_SCREENSHOT

screenshotId	Number	Identificador de la captura de pantalla recibida en el último análisis procesado	1680288125
--------------	--------	--	------------

Tabla 29: Modelado de datos para el asistente del profesor, usando el patrón "Single table design". Creación propia

En la Ilustración 48 se puede apreciar un ejemplo de cómo se guardan los datos en la tabla de dynamodb, siguiendo los patrones de acceso definidos a través de la llave de ordenamiento (Sort Key)

classId_pk	sk	attentionLevel	faceId	faceIndex	screenshotId
1680287994	LATEST_DISTRACTED#1680288125#9d0f4819-d6a6-4b2b-991f-890df88cfcac	44.33	9d0f4819-...	5	
1680287994	LATEST_DISTRACTED#1680288125#b4a25ff7-68d9-4caa-a378-fe9a9805fa17	29.53	b4a25ff7-6...	3	
1680287994	LATEST_DISTRACTED#1680288125#d8d00c05-35b0-4f73-a12d-4c3b3dbaaddf	61.7	d8d00c05-...	2	
1680287994	LATEST_SCREENSHOT				1680288125
1680274839	LATEST_SCREENSHOT				1680274842

Ilustración 48: Ejemplo de modelado de datos del asistente del profesor usando el patrón de "single table design". Creación propia

6.4.3.4. Obtener datos de análisis y rostros

Para poder mostrar los resultados de los análisis de manera visual se deben poder consultar los datos guardados de la manera más conveniente. Dado que los datos son guardados basados en el patrón de "Single-table-design", pueden usarse los patrones de acceso definidos usando la llave de ordenamiento para obtener los datos necesarios. En este caso, se crea la ruta "Obtener últimos rostros", el cual recibe un identificador de la clase (classId) y, haciendo uso de ambos patrones de acceso, busca la última captura de pantalla para la clase y obtiene los rostros y análisis de esa clase y captura de pantalla, luego responde con esta información para que la parte visual pueda mostrar adecuadamente esta información. En la Ilustración 49 se muestra un diagrama de flujo de esta función.

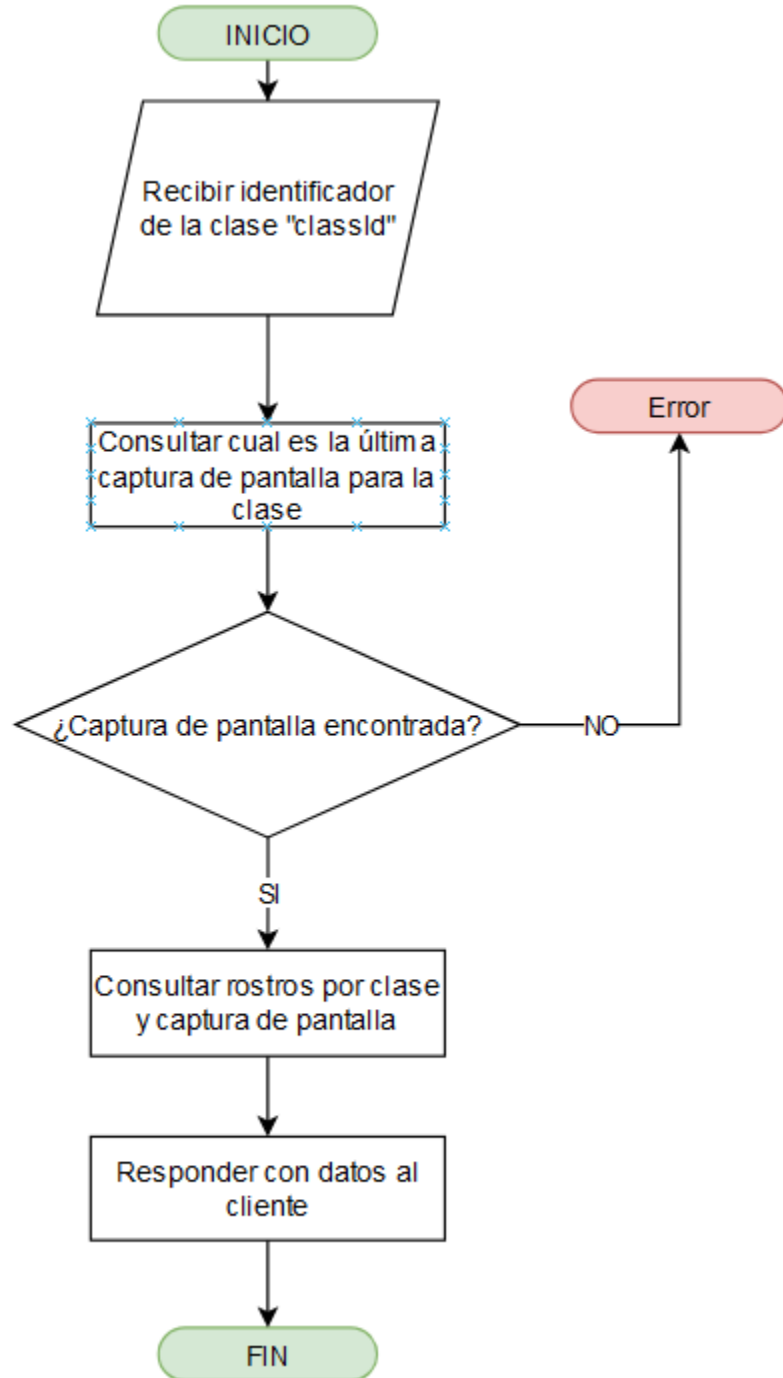


Ilustración 49: Diagrama de flujo de función "Obtener últimos rostros" del asistente del profesor. Creación propia

6.4.3.5. Generar nueva recomendación de actividad de aprendizaje activo

Para la generación de recomendaciones de actividad de aprendizaje activo, tal como se explicó en la sección 6.3.4 Generación de recomendaciones de actividades, para el alcance del prototipo se definieron dos tipos de actividades, las cuales pueden ser consultadas por el asistente de profesor (Frontend). En la Ilustración 35 se muestra el diagrama de flujo para generar dicha recomendación, junto con las protecciones y demás funcionalidades necesarias, como fue mostrado en la sección 6.3.4.2 Elección de sistema de procesamiento de lenguaje natural.

```
const activityTypeInput = getActivityTypeInput(activityType)

const studentsEducationLevelInput = getStudentsEducationLevelInput(studentsEducationLevel)
const classSubjectInput = getClassSubjectInput(classMainTopic, classSubTopic)

const isVirtualClassPrompt = `La clase es ${isVirtualClass ? 'virtual' : 'presencial'}.`

let inputPrompt = `${activityTypeInput.prompt}${studentsEducationLevelInput.prompt}${classSubjectInput.prompt}`

const responses = await openai.chatResponses(inputPrompt)

return {
  classId,
  input: {
    typeInput: activityTypeInput,
    subjectInput: classSubjectInput,
    studentsEducationLevelInput,
    isVirtualClass,
  },
  finalPrompt: inputPrompt,
  result: responses.map(response => response?.replaceAll('\n', '')),
  recommendedAt: getTimestamp(),
} as ClassActivityOutput
```

Ilustración 50: Fragmento de código del servicio de recomendación que genera una recomendación de actividad basada en el contexto de la clase. Creación propia

6.4.3.6. Infraestructura como código (IaC)

Al igual que se hizo en la sección 6.4.2.4 donde se definió las herramientas de infraestructura como código para el analizador de captura de pantalla, para el asistente del profesor se usa un enfoque similar, haciendo uso de Serverless Framework (Serverless Inc., 2023) y AWS Cloudformation (Amazon Web Services, 2023). En este caso, la herramienta Serverless Framework se encarga de generar la interacción necesaria para procesar eventos que son puestos en la cola en una función Lambda. También se encarga de configurar y generar la interacción necesaria entre las funciones lambda que sirven los datos al asistente del profesor (Frontend), y es capaz de crear y actualizar un “API Gateway” y conectarlo con sus

respectivas rutas y funciones Lambdas. Los otros recursos necesarios que deben crearse usando Cloudformation son:

- Tabla de DynamoDb para guardar y consultar los rostros y análisis procesados

En el anexo 1 se puede encontrar el vínculo al código fuente donde puede inspeccionarse con mayor detenimiento la infraestructura como código para el proyecto.

Al igual que en la sección 6.4.2.4, se define el rol que va a tener la función Lambda cuando se ejecuta. Dicho rol debe tener los permisos necesarios para comunicarse con los otros recursos y no tener permisos de más, tal como se muestra en la Ilustración 51.

```
// DynamoDB
{
  Effect: 'Allow',
  Action: [
    'dynamodb:List*',
    'dynamodb:BatchGet*',
    'dynamodb:DescribeTable',
    'dynamodb:Get*',
    'dynamodb:Query',
    'dynamodb:PutItem',
  ],
  Resource: [
    { "Fn::GetAtt": ["TeacherRecomendationTable", "Arn"] },
    `arn:aws:dynamodb:us-east-1:747016795213:table/classes-dev`,
    `arn:aws:dynamodb:us-east-1:747016795213:table/face-analysis-results-dev`
  ]
},
{
  Effect: 'Allow',
  Action: [
    'xray:PutTraceSegments',
    'xray:PutTelemetryRecords'
  ]
}
```

Ilustración 51: Fragmento de los permisos para el rol de la función Lambda. Creación propia

6.4.3.7. Interfaz de usuario

Para construir la interfaz de usuario, también conocido como “Frontend” del asistente virtual, se define usar React.js (Meta Open Source, 2023) para crear las vistas y Typescript como lenguaje de programación. También se decide construir los componentes usando “CSS Modules” (CSS Modules Open Source, 2023)

Para construir las vistas y siguiendo las buenas prácticas de React.js, se construyen componentes modulares que tengan un propósito definido y acotado. Luego, estos componentes pueden usarse en las páginas para generar las funcionalidades e interacciones apropiadas para el usuario.

Para la vista, se debe tener en cuenta que el usuario de esta interfaz de usuario es el profesor, y para que las recomendaciones de actividades de aprendizaje activo sean relevantes a la clase, el profesor debe poder ingresar esta información en la vista y dicha información debe enviarse en los llamados al servidor.

Se define una capa de servicios que se encarga de comunicarse con el servidor (Backend), ya sea para solicitar datos de análisis y rostros o para solicitar una nueva recomendación de actividad. Luego, siguiendo las buenas prácticas (Meta Open Source, 2023) se crea una capa de acceso usando “hooks” personalizados que permiten mantener el componente visual actualizado según se consulten los datos (o se genere y muestre un error, si es el caso) y, al mismo tiempo, mantiene la separación entre la lógica de llamados externos, la lógica de negocio y los componentes visuales.

```
async getLatestScreenshotFaces(classId: string): Promise<FaceAnalysis[]> {
  try {
    const latest = await api.get<GetLatestScreenshotFacesApiResponse>(
      `/latest-screenshot/${classId}`
    )

    console.log('latest', latest.latestFaces.length, latest.latestFaces)

    return latest && latest.latestFaces && Array.isArray(latest.latestFaces)
      ? latest.latestFaces.map(apiAnalysis => mapApiFaceAnalysisResultToFaceAnalysis(apiAnalysis))
      : []
  }
  catch (ex) {
    console.error('Error getting latest screenshot faces [service]', ex)
    throw ex
  }
}
```

Ilustración 52: Fragmento del código del servicio de análisis que se encarga de obtener los datos del servidor.
Creación propia

```

export default function useLatestScreenshotFaces() {
  const [latestScreenshotFaces, setLatestScreenshotFaces] = useState<FaceAnalysis[]>([])
  const [isLoading, setIsLoading] = useState(false)
  const [apiError, setApiError] = useState('')

  const getLatestScreenshotFaces = async (classId: string) => {
    try {
      setIsLoading(true)
      setApiError('')
      const latest = await AttentionService.getLatestScreenshotFaces(classId)
      if (latest) setLatestScreenshotFaces(latest)
    }
    catch (ex) {
      console.error('Error getting latest screenshot faces [hook]', ex)
      setApiError(JSON.stringify(ex))
    }
    setIsLoading(false)
  }

  const clearLatestScreenshotFaces = () => setLatestScreenshotFaces([])

  return {
    apiError,
    isLoading,
    latestScreenshotFaces,
    getLatestScreenshotFaces,
    clearLatestScreenshotFaces,
  }
}

```

Ilustración 53: Fragmento de código del "hook" personalizado para obtener el último análisis y rostros de este. Creación propia

```

export default function Button(props: ButtonProps) {
  const {
    children,
    onClick,
    colorVariant = ButtonColorVariants.PRIMARY,
    sizeVariant = ButtonSizeVariants.MEDIUM,
    disabled = false,
    style = {},
    className = '',
  } = props

  const btnClassName = `
    ${className}
    ${styles.btn}
    ${disabled ? styles.disabled : ''}
    ${!disabled && colorVariantClass(colorVariant)}
    ${sizeVariantClass(sizeVariant)}
  `

  return (
    <button
      className={btnClassName}
      style={style || {}}
      onClick={onClick}
      disabled={disabled}
    >
      {children}
    </button>
  )
}

```

Ilustración 54: Fragmento del código de un componente de botón de React reusable. Creación propia

```

<Button
  className={styles.btn}
  colorVariant={ButtonColorVariants.SECONDARY}
  sizeVariant={ButtonSizeVariants.LARGE}
  onClick={() => onStop()}
>
  Stop
  <FaRegEyeSlash className={styles.btnIcon} size="1.9rem"
</Button>

```

Ilustración 55: Fragmento del código que muestra el uso del componente reusable "botón" en una página. Creación propia

En las vistas creadas, se tuvo en cuenta las siguientes entradas del contexto de la clase:

- **Class ID:** Se refiere al identificador de la clase actual
- **Name:** Es el nombre del profesor que está usando el asistente
- **Class main topic:** Tema principal del que trata la clase o curso
- **Class sub topic:** Subtema que se está tratando en la clase, no es requerido
- **Education level:** Se refiere al nivel de educación de los alumnos, pueden ser alguno de los siguientes valores
 - **Primary:** Primaria
 - **Secondary:** Secundaria o bachillerato
 - **University:** Universitarios o pregrado
 - **Master:** Maestría
 - **Doctorate:** Doctorado

También se tiene un botón de inicio el cual solo se habilita cuando los campos requeridos son diligenciados. Para las vistas se tuvo en cuenta dos tipos de vistas: Vista de pantalla laptop (1024 pixeles de ancho o más) y vista de tableta (entre 680 y 1024 pixeles de ancho). Esto debido a que la herramienta está diseñada para usarse por un profesor mientras dicta una clase virtual, y por lo tanto debe tener una manera visual lo suficientemente grande para visualizar las ayudas del asistente.



Class ID * Name * Class main topic * Class sub topic Education level * Start

Class ID (Numbers only) Full Name Class main topic Class sub topic University Start



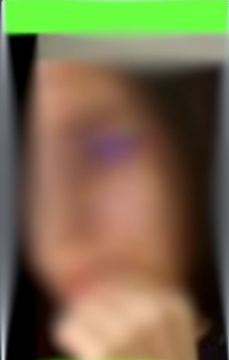


Please, fill the text boxes and hit the "Start class" button to start the assistant

Ilustración 56: Vista inicial de la interfaz de usuario del asistente del profesor en tamaño laptop. Creación propia

Class ID * 1680287994 Name * Lucas Vera Class main topic * Calculo diferencial

Class sub topic Limites Education level * University  Stop 

Last fetched at 00:30:02 GMT-0500 (Colombia Standard Time)

				
29.53 %	44.33 %	61.7 %	63.76 %	65


Recommend Activity 

Ilustración 57: Vista de la interfaz de usuario del asistente del profesor en tamaño laptop con el monitoreo de nivel de atención activado. Los rostros se han desenfocado para efectos de este documento. Creación propia

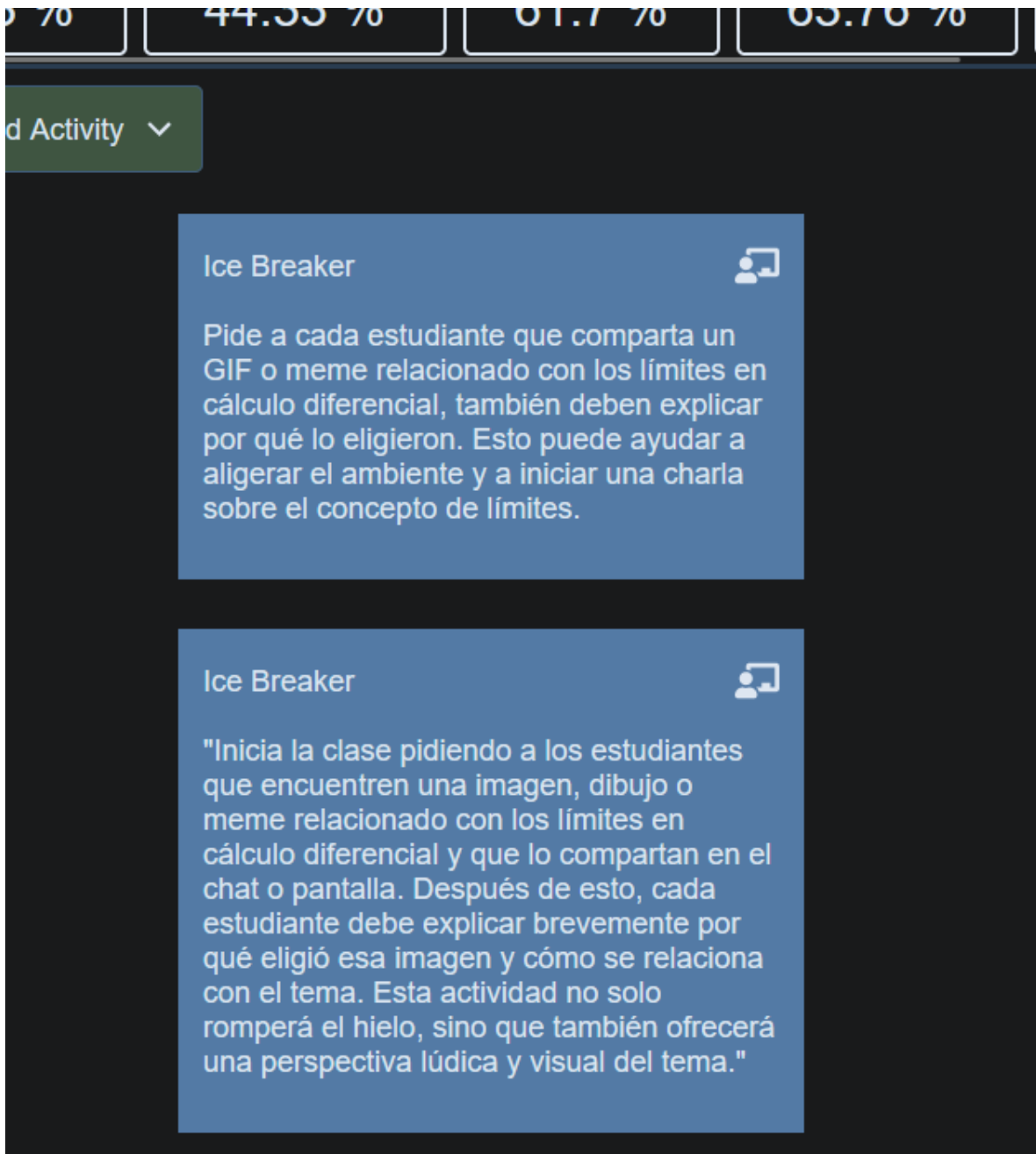


Ilustración 58: Ejemplos de actividades para "Romper el hielo" que se generan y muestran en la interfaz de usuario. Creación propia

6.4.4. Learning analytics

El alcance del prototipo en cuanto al módulo de learning analytics se acota a la captura y guardado de eventos en la base de datos de learning analytics. El componente visual diseñado en la sección 6.3.2.10 Diagrama de componentes “Learning Analytics Dashboard” que permite mostrar los datos recopilados por clase, profesor y alumnos, se propone para un posible trabajo futuro y se declara como fuera del alcance de este prototipo.

El módulo de learning analytics debe poder leer y procesar eventos enviados por otros sistemas, tales como nuevo análisis de nivel de atención, clase iniciada o finalizada. Para efectos del alcance del prototipo, solo se capturarán los eventos descritos. Los demás eventos descritos en la arquitectura de la solución en la sección 6.2.4 Learning analytics se proponen para futuras mejoras y quedan fuera del alcance de este prototipo.

6.4.4.1. Requisitos y tecnologías

Nombre	Descripción
Requisitos	<ul style="list-style-type: none">• Debe poder procesar nuevos eventos de resultados de análisis de rostros• Debe poder procesar nuevos eventos de cambios de estado de la clase, específicamente, eventos de cuando la clase comienza y cuando termina• Debe poder almacenar datos relacionado a los eventos que procesa• Debe poder almacenar diferentes tipos de entidades y relacionarlas entre si
Elección de tecnologías	<ul style="list-style-type: none">• Lenguaje de programación: Typescript• Entorno de ejecución: AWS Node.js Lambda• Herramienta para interactuar con la base de datos: Sequelize• Herramienta de despliegue: Serverless Framework• Herramienta de infraestructura como código: Serverless Framework y Cloudformation• Editor de código seleccionado: vscode• Base de datos seleccionada: AWS RDS (Postgres)

Tabla 30: Requisitos y tecnologías elegidas para la implementación del módulo de learning analytics en el prototipo. Creación propia

6.4.4.2. Procesamiento de resultados de análisis

Se define un enfoque similar a lo explicado en la sección 6.4.3.2 Procesamiento de resultados de análisis perteneciente al asistente del profesor. Se hace uso de la configuración con Serverless Framework para vincular la cola de learning analytics con una función Lambda que permita procesar los mensajes a medida que se reciban, de acuerdo con el patrón “Fan-Out” (Richardson & Fröhner, 2019). En la Ilustración 46 de la sección del asistente del profesor se muestra un fragmento de la configuración necesaria para dicho fin, la cual sería idéntica a la necesaria en el módulo de learning analytics.

En cuanto al flujo de los datos y la lógica, en el módulo de asistente del profesor es necesario consultar la base de datos de los resultados del análisis para obtener todos los detalles necesarios para suplir su función. En el caso del módulo de learning analytics, basta con realizar un guardado en la propia base de datos solo con la información identificada del mensaje.

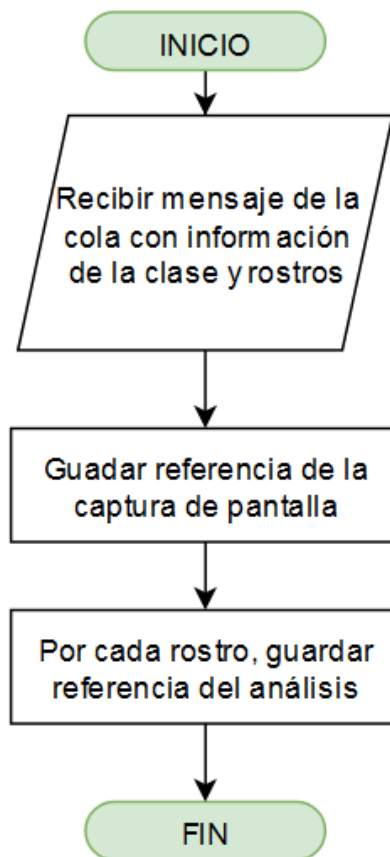


Ilustración 59: Diagrama de flujo de la lógica para procesar eventos de resultados de análisis. Creación propia

6.4.4.3. Modelado de datos y almacenamiento

Al procesar los eventos obtenidos, se deben guardar datos por entidades y dichas entidades deben estar relacionadas de manera jerárquica. Las relaciones entre las entidades guardadas en base de datos deben ser lo suficientemente flexible para que se puedan realizar diferentes consultas sobre los datos, tales como aplicar filtros, agrupamientos y ordenamientos. Es por lo anterior que se define la elección de una base de datos relacional, dado que dichas bases de datos permiten definir entidades y relacionar entidades entre sí, al mismo tiempo, creando flexibilidad en las consultas y posibilidad de extender (Date, 2015). En la Ilustración 60 se puede apreciar el diagrama entidad-relación para la base de datos de learning analytics. Para cumplir la funcionalidad deseada en este prototipo, se definen las siguientes entidades:

- **Classes:** se define como la entidad “Clase” y contiene información del contexto de la clase como el nombre del profesor, el tema y subtema de la clase, y el nivel de educación de los alumnos. Su llave primaria es *id*, el cual es el identificador de la clase que se reportó por el resultado de análisis y la aplicación de toma de captura de pantalla.
- **ClassScreenshots:** Es la entidad que representa las capturas de pantalla de las clases y contiene la información de las capturas de pantalla junto con información del análisis identificado en dicha captura de pantalla. Su identificador es *id*, el cual identifica la captura de pantalla. Tiene una llave foránea (FK), *classid*, la cual se relaciona con la tabla de “Classes” a través de su *id*.
- **ClassFaces:** Es la entidad que representa los rostros de la clase, por captura de pantalla. Contiene información del análisis específico para cada rostro. Su identificador es *id*, el cual identifica esta entidad de manera única. Tiene dos llaves foráneas (FK), *classid* y *screenshotid*. Ambas llaves foráneas se usan para relacionar las tablas de clases y de capturas de pantalla, respectivamente.

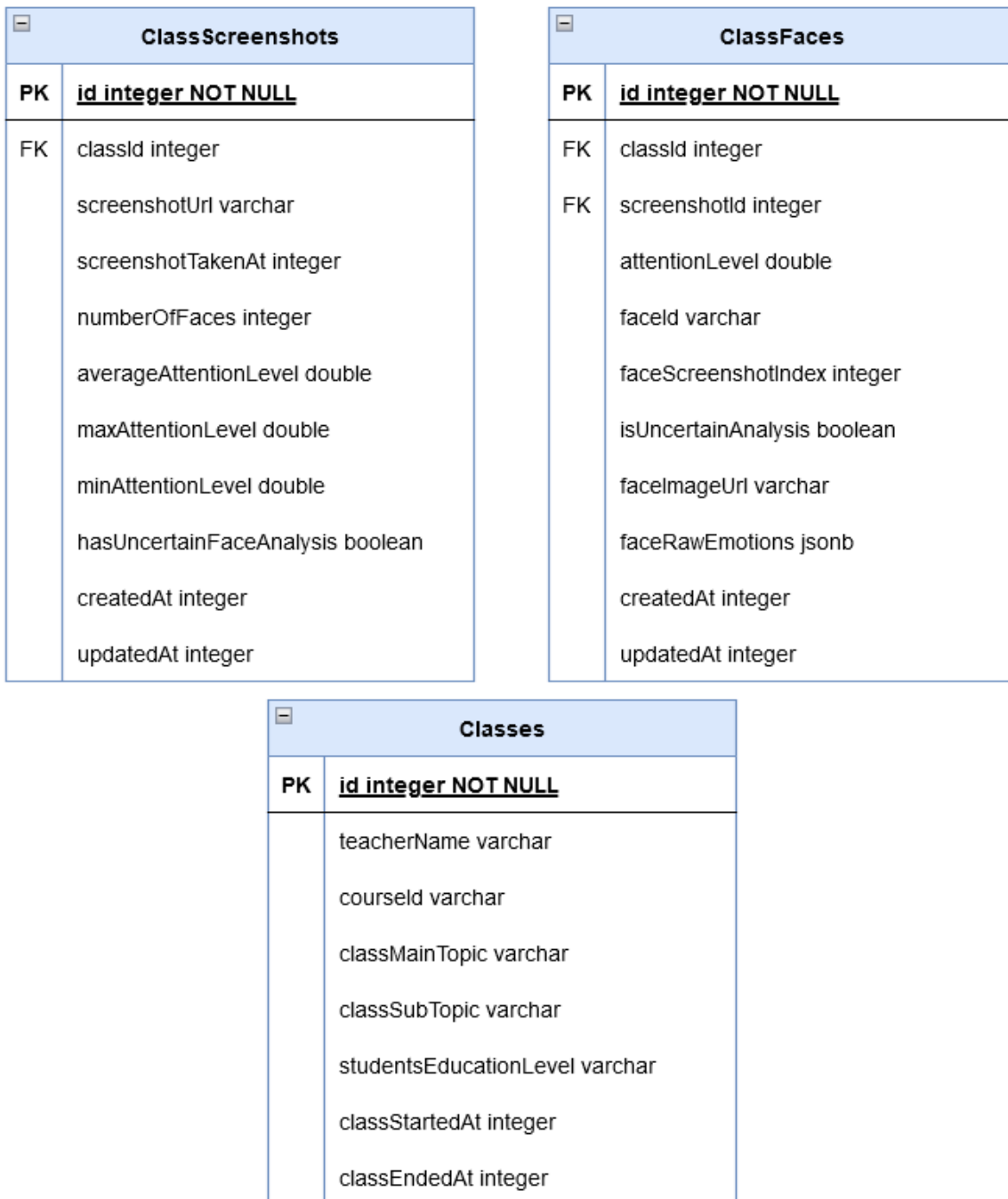


Ilustración 60: Diagrama de entidad-relación para la base de datos de learning analytics. Creación propia

Tabla: Classes			
Nombre	Tipo de dato	Descripción	Ejemplo
id (PK)	integer	Identificador de la clase. Es la llave primaria sobre la cual se identifica cada fila. Requerido	1680287994
teacherName	varchar (255)	Nombre completo del profesor que dicta la clase, tal como se reporta desde el asistente	Lucas Vera
courseId	varchar (255)	Identificador del curso al que pertenece la clase. Por ahora se deja en blanco para posible mejora en el futuro	CAL-152
mainTopic	varchar (255)	Tema principal de la clase o curso	Calculo diferencial
subTopic	varchar (255)	Subtema de la clase	Limites
studentsEducationLevel	varchar (255)	Nivel de atención de los estudiantes de la clase	University
startedAt	integer	Momento en que la clase inició, en formato Unix timestamp	1680287994
endedAt	integer	Momento en que la clase finalizó, en format Unix timestamp	1680287994

Tabla 31: Modelado para la entidad de clase en la tabla "Classes". Creación propia

Tabla: ClassScreenshots			
Nombre	Tipo de dato	Descripción	Ejemplo
id (PK)	integer	Identificador de la captura de pantalla. Es la llave primaria sobre la cual se identifica cada fila. Requerido	1680287995
classId (FK)	integer	Identificador de la clase a la que pertenece la captura de pantalla. Requerido	1680287994
screenshotUrl	varchar (255)	Vínculo de la captura de pantalla tal como fue tomada	https://class-screenshot.s3.amazonaws.com/classes/2023-3-31/classid-1680274839/screenshotid-1680274842.jpg
screenshotTakenAt	integer	Momento en que fue tomada la captura de pantalla, en formato Unix timestamp	1680287994
averageAttentionLevel	double	Representa el valor promedio de nivel de atención para la captura de pantalla	42.63
maxAttentionLevel	double	Representa el mayor nivel de atención encontrado en la captura de pantalla	57.16
minAttentionLevel	double	Representa el menor nivel de atención encontrado en la captura de pantalla	21.46
hasUncertainFaceAnalysis	boolean	Indica si alguno de los rostros presentes en la captura de pantalla es marcado como "análisis incierto"	false

Tabla 32: Modelado para la entidad de captura de pantalla en la tabla "ClassScreenshots". Creación propia

Tabla: ClassFaces			
Nombre	Tipo de dato	Descripción	Ejemplo
id (PK)	integer	Identificador del rostro. Es la llave primaria sobre la cual se identifica cada fila. Requerido	1680287996
classId (FK)	integer	Identificador de la clase a la que pertenecen los rostros. Requerido	1680287994
screenshotId (FK)	integer	Identificador de la captura de pantalla al que pertenecen los rostros. Requerido	1680287995
attentionLevel	double	Valor del nivel de atención calculado para el rostro	64.51
faceId	varchar (255)	Identificador del rostro generado por AWS Rekognition	91a96054-56f6-4682-959f-173472db7043
faceScreenshotIndex	integer	Índice del rostro dentro del análisis e identificación de rostros en la captura de pantalla	2
isUncertainAnalysis	boolean	Indica si el análisis del rostro es incierto para esa captura de pantalla	false
faceImageUrl	varchar (255)	Vínculo donde se encuentra alojado el recorte del rostro para la captura de pantalla	https://classesdev.s3.amazonaws.com/classes/classid-1680287994/screenshotid-1680288035/faces/faceindex-1.jpg
faceRawEmotions	jsonb	Esta columna almacena las emociones tal como fueron identificadas en el rostro por AWS Rekognition	[{"SAD":93.3779525756836}, {"CALM":32.189117431640625}, {"CONFUSED":18.225250244140625}, ...

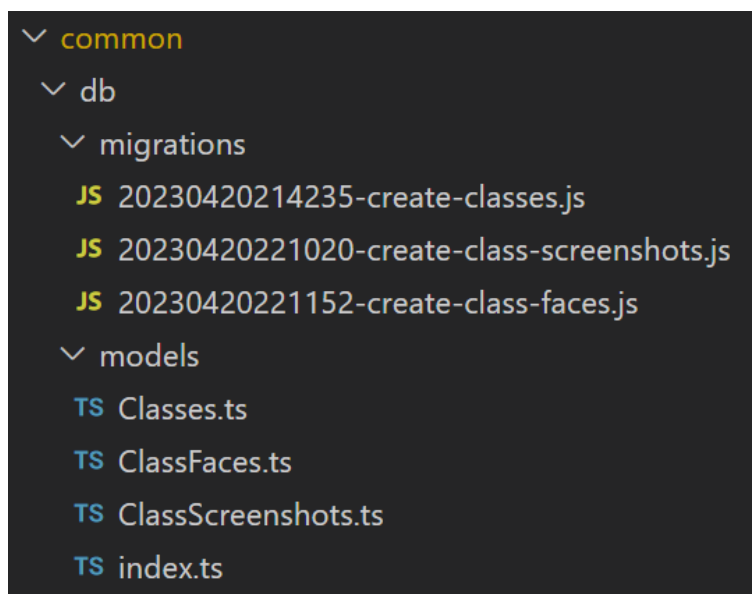
Tabla 33: Modelado para la entidad de rostro en la tabla "ClassFaces". Creación propia

6.4.4.4. Gestión de la base de datos a través de un ORM

Existen diversas opciones para interactuar con una base de datos relacional en una aplicación. Una manera es a través de una herramienta de mapeo objeto-relacional (ORM, “object-relational mapping” por su nombre en inglés). Este enfoque permite interactuar con la base de datos de manera orientada a objetos, y se encarga de manejar los detalles de interacción con el motor de SQL sin necesidad de exponer dicha complejidad en el código. Esto permite que el código sea más fácil de escribir y entender para un desarrollador (Fowler, 2003).

El ORM elegido es sequelize, el cual es un ORM diseñado para el ecosistema de node.js. El paquete “sequelize-typescript” (Sequelize, 2023) permite integrarse de manera nativa con typescript, el cual es el lenguaje usado para desarrollar la aplicación.

Al pensar en un diseño de base de datos que sea evolutivo, se encuentra que los cambios en el esquema y relaciones pueden ser frecuentes a medida que se introducen características o el proyecto evoluciona de manera natural. Es por esto que se introduce el concepto de “Migraciones”, las cuales son archivos tipo “script” que permiten alterar el diseño de la base de datos (Sequelize, 2023). Bajo un diseño evolutivo de base de datos, las migraciones se vuelven cruciales para mantener un versionamiento claro de los cambios en el esquema y diseño, permite replicar la base de datos más fácilmente en diferentes entornos, permite tener la base de datos al mismo nivel del código, entre otros beneficios (Sadalage & Fowler, 2016).



```

  ✓ common
    ✓ db
      ✓ migrations
        JS 20230420214235-create-classes.js
        JS 20230420221020-create-class-screenshots.js
        JS 20230420221152-create-class-faces.js
      ✓ models
        TS Classes.ts
        TS ClassFaces.ts
        TS ClassScreenshots.ts
        TS index.ts

```

Ilustración 61: Estructura de las migraciones y modelos implementados usando Sequelize. Creación propia.

```

module.exports = {
  async up (queryInterface, Sequelize) {
    await queryInterface.createTable('ClassFaces', {
      id: {
        allowNull: false,
        primaryKey: true,
        type: Sequelize.INTEGER
      },
      classId: {
        type: Sequelize.INTEGER,
        allowNull: false,
      },
      screenshotId: {
        type: Sequelize.INTEGER,
        allowNull: false,
      },
      attentionLevel: {
        type: Sequelize.DOUBLE
      },
      faceId: {
        type: Sequelize.STRING
      },
      faceScreenshotIndex: {
        type: Sequelize.INTEGER
      },
      isUncertainAnalysis: {
        type: Sequelize.BOOLEAN
      },
      faceImageUrl: {
        type: Sequelize.STRING
      },
      faceRawEmotions: {
        type: Sequelize.JSONB
      }
    });
  }
};

```

Ilustración 62: Fragmento de código para la migración de la tabla "ClassFaces". Creación propia.

```

export interface ClassFaceProperties {
  id: number,
  classId: number,
  screenshotId: number,
  attentionLevel: number,
  faceId: string,
  faceScreenshotIndex: number,
  isUncertainAnalysis: boolean,
  faceImageUrl: string,
  faceRawEmotions: string,
  analysisCreatedAt: number,
}

export class ClassFaces extends Model implements ClassFaceProperties {
  @PrimaryKey
  @AutoIncrement
  @Column(DataType.INTEGER)
  id: number

  @ForeignKey(() => Classes)
  @Column(DataType.INTEGER)
  classId: number

  @BelongsTo(() => Classes)
  class: Classes

  @ForeignKey(() => ClassScreenshots)
  @Column(DataType.INTEGER)
  screenshotId: number

  @BelongsTo(() => ClassScreenshots)
  classScreenshot: ClassScreenshots

  @Column(DataType.INTEGER)
  attentionLevel: number

  @Column(DataType.STRING)
  faceId: string

  @Column(DataType.INTEGER)
  faceScreenshotIndex: number
}

```

Ilustración 63: Fragmento del código que muestra el objeto-relación para la entidad rostro de clase y la tabla "ClassFaces" usando sequelize-typescript. Se puede apreciar las relaciones con otras entidades. Creación propia.

```

const createPromise = ClassFaces.create({
  classId,
  screenshotId,
  attentionLevel,
  faceId,
  faceScreenshotIndex: index,
  isUncertainAnalysis: isUncertainAnalysis,
  faceImageUrl: faceS3Url,
  faceRawEmotions: JSON.stringify(face.Emotions),
  analysisCreatedAt,
})

return createPromise
)

```

Ilustración 64: Fragmento del código que muestra cómo se guarda un rostro en base de datos usando las clases del ORM. Creación propia

6.4.4.5. Infraestructura como código

De igual forma que se definió en secciones anteriores (6.4.3.6 y 6.4.2.4), para definir la infraestructura como código se hace uso de Serverless Framework para generar la configuración necesaria para consumir los mensajes puestos en la cola. Para crear la base de datos se hace uso de cloudformation. De acuerdo con las practicas recomendadas (Amazon Web Services, 2023), se crea la base de datos dentro de su propio espacio en la nube, conocido como VPC. Esto mejora la seguridad al aislar la entrada a la base de datos solo por medio de la compuerta de internet, subnets y grupo de seguridad.

- Base de datos de learning analytics
 - Instancia de RDS usando postgres
 - Grupo de Subnets para la instancia RDS
- Nube privada virtual (VPC) para alojar la base de datos de manera segura
 - Configuración de VPC
 - Subnets públicas para la instancia RDS
 - Grupo de seguridad (security group)
 - Compuerta de internet (Internet Gateway)
 - Unión de compuerta con VPC
 - Tabla de enrutamiento y rutas
 - Asociación de tabla de enrutamiento con subnets

Dada la naturaleza de “prototipo” de este proyecto, se considera que la familia de la instancia de la base de datos se encuentre dentro del nivel gratuito de AWS, para no incurrir en costos mientras se construye y prueba el prototipo. Una vez se tenga mayor tracción, se puede realizar un escalamiento de la instancia para que tenga mayor almacenamiento y procesamiento. Siguiendo el mismo enfoque, no se configuran copias de seguridad automatizadas, replicación en otras regiones o estrategias de cacheo.

```
RDSInstance: {
  Type: "AWS::RDS::DBInstance",
  Properties: {
    Engine: "postgres",
    DBInstanceIdentifier: `learninganalytics${stage}`,
    DBName: `${DbBaseName}${stage}`,
    MasterUsername,
    MasterUserPassword,
    AllocatedStorage: 20, // free tier
    DBInstanceClass: "db.t4g.micro", // free tier
    VPCSecurityGroups: [
      {
        Ref: "SecurityGroup"
      }
    ],
    DBSubnetGroupName: {
      Ref: "DBSubnetGroup"
    },
    PubliclyAccessible: true
  }
},
DBSubnetGroup: {
  Type: "AWS::RDS::DBSubnetGroup",
  Properties: {
    DBSubnetGroupDescription: `Subnet group for learning analytics rds instance in ${stage}`,
    SubnetIds: [
      {
        Ref: "PublicSubnet1"
      },
      {
        Ref: "PublicSubnet2"
      }
    ]
  }
}
```

Ilustración 65: Fragmento del código para crear y configurar una instancia de base de datos AWS RDS postgres. Creación propia

```

VPC: {
  Type: "AWS::EC2::VPC",
  Properties: {
    CidrBlock: "10.0.0.0/16",
    EnableDnsSupport: true,
    EnableDnsHostnames: true,
    Tags: [
      {
        Key: "Name",
        Value: `${learningAnalyticsName} rds vpc`
      }
    ]
  }
},
PublicSubnet1: {
  Type: "AWS::EC2::Subnet",
  Properties: {
    VpcId: {
      Ref: "VPC"
    },
    CidrBlock: "10.0.1.0/24",
    AvailabilityZone: {
      "Fn::Select": [
        0,
        {
          "Fn::GetAZs": ""
        }
      ]
    },
    Tags: [
      {
        Key: "Name",
        Value: `${learningAnalyticsName} rds public subnet 1`
      }
    ]
  }
},
PublicSubnet2: {
  Type: "AWS::EC2::Subnet",

```

Ilustración 66: Fragmento de la infraestructura como código para crear una VPC y configurar la base de datos dentro. Creación propia

7. CONCLUSIONES Y TRABAJOS FUTUROS

En esta sección se presentan los enunciados finales sobre el trabajo realizado y las apreciaciones obtenidas del documento. También se muestran propuestas para continuar explorando esta área del conocimiento a través de mejoras y pruebas detalladas.

7.1. RESULTADOS Y APRECIACIONES FINALES

De acuerdo con el análisis realizado sobre los desafíos que enfrentan los docentes en la medición no intrusiva del nivel de atención de los estudiantes, se encuentra que este reto es especialmente importante en clases virtuales. Los docentes que dictan una clase de manera presencial tienen una mejor visión del nivel de atención de sus estudiantes, por lo que pueden incorporar técnicas de aprendizaje activo en sus clases de manera más acertada que un profesor que dicte una clase virtual.

En cuanto a calcular el nivel de atención de los estudiantes en una clase virtual, existen varios métodos que se han explorado para este fin. Este trabajo se enfocó en el expuesto por (Tabassum, Allen, & De, 2020), en el cual los autores generaron un modelo de aprendizaje de máquina basado en datos recolectados por ellos mismos de estudiantes recibiendo clases, luego hacen una correlación entre los resultados de su modelo para identificar nivel de atención en estudiantes y las emociones identificadas en rostros por “AWS Rekognition”. Este trabajo codificó esta relación para ser usada como recurso para determinar el nivel de atención de los estudiantes en una clase virtual.

Este trabajo abarcó el diseño de una arquitectura de una solución digital que fuera capaz de medir el nivel de atención de los estudiantes en una clase virtual, de manera no intrusiva. Que dicho nivel de atención pudiera visualizarse en una herramienta de apoyo al docente. Esta herramienta de apoyo al docente facilita la implementación de técnicas de aprendizaje activo al no solo mostrar al docente el estado de la clase y los alumnos en cuanto a su nivel de atención, sino que también puede usar la herramienta para generar una recomendación de actividad. Dicha recomendación de actividad se genera usando un modelo de procesamiento de lenguaje natural (NLP) para que dicha recomendación sea acorde al contexto de la clase: tema de la clase, nivel de educación de estudiantes, entre otros. Además, dicha solución abarca un módulo de analítica enfocada al aprendizaje (Learning analytics), donde dicho módulo busca almacenar y mostrar el rendimiento de las clases y alumnos en cuanto a su nivel de atención de manera histórica, permitiendo a los docentes analizar las variaciones en el nivel de atención de sus alumnos, y analizar como las diferentes actividades impactan el nivel de atención de la clase.

La arquitectura usada para la construcción de la solución se basó en tecnologías desacopladas en microservicios, en la nube y “serverless”. El uso de estas

tecnologías permite que los sistemas creados gocen de escalamiento automático, reducción de costos, desacoplamiento y separación de intereses, alta disponibilidad y buena velocidad en desarrollo y despliegues. Por otro lado, los compromisos asumidos al usar estas tecnologías tienen que ver con incremento de complejidad en observabilidad, latencia debido a “arranque en frío” (cold start), límite de tiempo de ejecución y bloqueo de vendedor (vendor lock-in).

El prototipo construido representa una oportunidad valiosa para mejorar el proceso de enseñanza de los docentes cuando dictan clases virtuales, en específico por los siguientes puntos:

- Genera una medición del nivel de atención de los estudiantes de manera no intrusiva, lo que evita interrumpir la clase o generar distracciones innecesarias que no contribuyen a la clase.
- Presenta este nivel de atención de manera visual y fácil de entender para un docente, el cual puede apoyarse en este material para entender el estado actual de la clase y tomar acción para mejorar el aprendizaje de los alumnos si lo considera pertinente.
- Permite generar recomendaciones de actividades de aprendizaje activo flexibles y que son de utilidad para las clases. Esto ya que usa técnicas de generación de texto apoyados en modelos de procesamiento de lenguaje natural de última generación, lo cual permite generar recomendaciones que son relevantes y tienen coherencia con el contexto de la clase y sus participantes.
- Recopila datos históricos a lo largo de cada clase, lo que permite realizar un análisis posterior a las clases donde cada docente pueda evaluar el desempeño de cada clase basado en el nivel de atención de las clases a lo largo del tiempo, y de sus alumnos.
- La arquitectura técnica seleccionada habilita al prototipo para ser usado por diferentes instituciones educativas y clases de manera simultánea, sin que esto comprometa su precisión, rendimiento o seguridad.

7.2. TRABAJOS FUTUROS

En trabajos futuros se propone continuar con el desarrollo del prototipo, en específico, expandir las recomendaciones de actividades de aprendizaje activo y validarlas con experimentos en clases reales. En específico, se resaltan los siguientes puntos:

- Como posible trabajo futuro se propone la verificación del impacto del prototipo sobre el proceso de aprendizaje de los estudiantes que participen de clases virtuales con educadores que hagan uso de el asistente desarrollado.

- Mejorar el módulo de “learning analytics” para que tenga un alcance mayor, es decir, que almacene más datos y las visualizaciones en el panel tengan una mayor cobertura que pueda usarse para analizar más a fondo comportamiento por estudiante, por docente, por curso o, inclusive, por institución educativa.
- En cuanto a la captura del nivel de atención, se propone realizar experimentos con diferentes implementaciones de este cálculo, basándose en otras estrategias para esta computación y evaluar la precisión que presenten con el fin de generar una comparativa de los diferentes métodos disponibles.
- Se propone también como posible trabajo futuro, tener una integración más directa con la plataforma donde se dicta la clase virtual (Microsoft Teams), facilitando que la herramienta interactúe con los participantes de la clase de manera más directa y liberando al educador de coordinar las actividades de aprendizaje activo.
- En el futuro, la herramienta podría verse beneficiada de tener un módulo para el alumno, de manera que los alumnos puedan recibir retroalimentación a medida que transcurre la clase y al finalizar la clase pueda inspeccionar los datos pertinentes.
- El prototipo creado se generó para clases virtuales síncronas, sin embargo, se propone en el futuro inspeccionar la utilidad de esta estrategia en ambientes de educación presenciales como clases en universidades o colegios, conferencias, charlas, discursos, u otro tipo de escenarios donde se encuentren uno o más oradores y una audiencia.

8. REFERENCIAS

Amazon Web Services. (2021, Octubre 21). *Deploy and host app*. Retrieved from Amplify Dev Center: <https://docs.amplify.aws/start/getting-started/hosting/q/integration/js/>

Amazon Web Services. (2023). *AWS Cloudformation Walkthrough: Updating a stack*. Retrieved from AWS Cloudformation User Guide: <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/updating.stacks.walkthrough.html>

Amazon Web Services. (2023). *AWS Lambda Pricing*. Retrieved from AWS Lambda: <https://aws.amazon.com/lambda/pricing/>

Amazon Web Services. (2023). *AWS Service Level Agreements (SLAs)*. Retrieved from AWS: <https://aws.amazon.com/legal/service-level-agreements/>

Amazon Web Services. (2023). *Best practices for Amazon RDS*. Retrieved from AWS RDS Documentation: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.html#CHAP_BestPractices.DiskPerformance

Amazon Web Services. (2023). *Bounding Box*. Retrieved from Amazon Rekognition API Documentation: https://docs.aws.amazon.com/rekognition/latest/APIReference/API_BoundingBox.html

Amazon Web Services. (2023). *DetectFaces - AWS API Documentation*. Retrieved from Amazon Rekognition API Documentation: https://docs.aws.amazon.com/rekognition/latest/APIReference/API_DetectFaces.html

Amazon Web Services. (2023). *Emotion - AWS API Docs*. Retrieved from Amazon Rekognition Documentation: https://docs.aws.amazon.com/rekognition/latest/APIReference/API_Emotion.html

Amazon Web Services. (2023). *Integration AWS X-Ray with other AWS services*. Retrieved from AWS Developer Guide: <https://docs.aws.amazon.com/xray/latest/devguide/xray-services.html>

Amazon Web Services. (2023). *Searching faces in a collection*. Retrieved from Amazon Rekognition Developer Guide: <https://docs.aws.amazon.com/rekognition/latest/dg/collections.html>

Amazon Web Services. (2023). *What is Amazon CloudWatch*. Retrieved from AWS Documentation User Guide: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>

Amazon Web Services. (2023). *What's the Difference Between Monolithic and Microservices Architecture?* Retrieved from Amazon Web Services: <https://aws.amazon.com/compare/the-difference-between-monolithic-and-microservices-architecture/>

Amazon Web Services. (2023). *Working with images*. Retrieved from Amazon Rekognition Documentation: <https://docs.aws.amazon.com/rekognition/latest/dg/images.html>

Bala, R., Smith, D., Ji, K., Wright, D., & Borrega, M. A. (2022, October 19). *Magic Quadrant for Cloud Infrastructure and Platform Services*. Retrieved from Gartner: <https://www.gartner.com/doc/reprints?id=1-2AOZQAQL&ct=220728&st=sb>

Barton Essel, H., Vlachopoulos, D., Tachie-Menson, A., Eduafua Johnson, E., & Kwame Baah, P. (2022). The impact of a virtual teaching assistant (chatbot)

on students' learning in Ghanaian higher education. *International Journal of Educational Technology in Higher Education*, 1-19.

Beswick, J. (2021, April 26). *Operating Lambda: Performance optimization – Part 1*. Retrieved from AWS Compute Blog: <https://aws.amazon.com/blogs/compute/operating-lambda-performance-optimization-part-1/>

Bonwell, C. C., & Eison, J. A. (1991). Active Learning: Creating Excitement in the Classroom. *ASHE-ERIC Higher Education Report*, 1-121.

Borup, J., West, R. E., & Graham, C. R. (2012). Improving online social presence through asynchronous video. *Internet and Higher Education* 15, 195-203.

Brown, S. (2023). *The C4 model for visualising software architecture*. Retrieved from C4 Model: <https://c4model.com/>

Cavanagh, M. (2011). Students' experiences of active engagement through cooperative learning activities in lectures. *Active Learning in Higher Education* 12, 23-33.

Cheng, P.-Y., Chien, Y.-C., & Huang, Y.-M. (2017). The Design and Implementation of a Real-Time Attention Recognition/Feedback System in Online Learning Course. *The Sixth International Conference of Educational Innovation through Technology*, 214-217.

Chi, M. T., & Wylie, R. (2014). The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 219-243.

Chung, L., & Sampaio do Prado Leite, J. C. (2009). On Non-Functional Requirements in Software Engineering. *Mylopoulos Festschrift, LNCS 5600*, 363-379.

Coalition for Psychology in Schools and Education. (2020, September 29). *Managing attention and distractibility in online learning*. Retrieved from American

Psychological Association: <https://www.apa.org/topics/covid-19/managing-attention-distractibility-online-learning>

Cohn, M. (2004). *User Stories Applied*. Addison-Wesley Professional.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *arXiv*, 1-47.

Crawford, J. A., Jurgen, R., Butler-Henderson, K. A., & Malkawi, B. H. (2020). COVID-19: 20 countries' higher education intra-period digital pedagogy responses. *Journal of Applied Learning and Teaching*, 1-20.

CSS Modules Open Source. (2023). *css-modules*. Retrieved from Github: <https://github.com/css-modules/css-modules>

Cui, Y. (2020, Septiembre 20). *AWS Lambda Cold Starts: Solving the Problem*. Retrieved from Lumigo: <https://lumigo.io/blog/this-is-all-you-need-to-know-about-lambda-cold-starts/>

Dale, R. (2021). GPT-3: What's it good for? *Natural Language Engineering*, 113-118.

Date, C. (2015). *SQL And Relational Theory*. Sebastopol: O'Reilly.

De Rooms, B. (2020, Febrero 6). *A Comparison of Serverless Providers*. Retrieved from Fauna: <https://fauna.com/blog/comparison-faas-providers>

DeBrie, A. (2020, Febrero 5). *The What, Why, and When of Single-Table Design with DynamoDB*. Retrieved from DeBrie Advisory: <https://www.alexdebrie.com/posts/dynamodb-single-table/>

Dixson, M. D. (2010). Creating effective student engagement in online courses: What do students find engaging? *Journal of the Scholarship of Teaching and Learning* 10, 1-13.

Edel-Malizia, S. (2009). *Teaching with Video*. Retrieved from PennState: <https://sites.psu.edu/pedagogicalpractices/teaching-with-video/>

- Essel, H. B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E. E., & Baah, P. K. (2022). The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education. *International Journal of Educational Technology in Higher Education*, 1-19.
- Evans, B. (2023). *screenshot-desktop*. Retrieved from Github: <https://github.com/bencevans/screenshot-desktop#readme>
- Fee, N. (2020, August 28). *What Is Serverless Architecture? Key Benefits and Limitations*. Retrieved from New Relic Blog: <https://newrelic.com/blog/best-practices/what-is-serverless-architecture>
- Fisher, C. (2018). Cloud versus On-Premise Computing. *American Journal of Industrial and Business Management*, 1991-2006.
- Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.
- Gamrat, C., Lenze, L., Bardzell, J., & Glantz, E. (2021, March 16). *Improved Student Engagement in Higher Education's Next Normal*. Retrieved from EDUCAUSE: <https://er.educause.edu/articles/2021/3/improved-student-engagement-in-higher-educations-next-normal>
- Google Cloud Architecture Center. (2022, October 28). *Patterns for scalable and resilient apps*. Retrieved from Google Cloud Architecture Center: <https://cloud.google.com/architecture/scalable-and-resilient-apps>
- Hazdun, N. (2023, May 27). *Cloud Versus On Premises: Advantages And Disadvantages Of Both Models*. Retrieved from Forbes Technology Council: <https://www.forbes.com/sites/forbestechcouncil/2023/03/27/cloud-versus-on-premises-advantages-and-disadvantages-of-both-models/?sh=2755b9c35df6>
- IBM. (2023). *What is serverless?* Retrieved from IBM topics: <https://www.ibm.com/topics/serverless>

- IBM Market Development & Insights. (2021). *Serverless in the enterprise, 2021: Building the next generation of efficient, flexible, cost-effective cloud native applications*. IBM.
- IEEE and the Open Group. (2018). *IEEE POSIX.1-2017 General Concepts*. Retrieved from Open Group Standard: https://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html
- Janiesch, C., Zschech, P., & Henrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 685-695.
- Jet Brains. (2022). *The State of Developer Ecosystem 2022*. Retrieved from Jet Brains: <https://www.jetbrains.com/lp/devecosystem-2022/>
- Key CDN. (2023, 5 5). *Load Balancing*. Retrieved from KeyCDN Support: <https://www.keycdn.com/support/load-balancing>
- Larman, C. (2004). System Sequence Diagrams. In C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition* (pp. 252-265). New Jersey: Addison Wesley Professional.
- Lassa, H. (2019, Diciembre 10). *AWS re:Invent 2019: [REPEAT 3] Serverless architectural patterns and best practices (ARC307-R3)*. Retrieved from Youtube [Video]: <https://www.youtube.com/watch?v=9IYpGTS7Jy0>
- Lefèvre, X. (2020, May 19). *What a typical 100% Serverless Architecture looks like in AWS*. Retrieved from Medium: <https://medium.com/serverless-transformation/what-a-typical-100-serverless-architecture-looks-like-in-aws-40f252cd0ecb>
- Lovell, F., & Others. (2023). *Sharp*. Retrieved from Github: <https://github.com/lovel/sharp>

- Madsen, J., Júlio, S. U., Gucik, P. J., Steinberg, R., & Parra, C. L. (2020). Synchronized eye movements predict test scores in online video education. *Psychological and Cognitive Sciences*, 1-9.
- Mangaroska, K., & Giannakos, M. (2019). Learning Analytics for Learning Design: A Systematic Literature Review of Analytics-Driven Design to Enhance Learning. *IEEE Transactions On Learning Technologies* 12, 516-534.
- Meta Open Source. (2023). *React Documentation*. Retrieved from React: <https://react.dev/learn>
- Meta Open Source. (2023). *Reusing Logic with Custom Hooks*. Retrieved from React Documentation: <https://react.dev/learn/reusing-logic-with-custom-hooks#extracting-your-own-custom-hook-from-a-component>
- Mezzalira, L., Hyatt, L., Denti, V., & Jaupaj, Z. (2022, Mayo 2022). *Let's Architect! Serverless architecture on AWS*. Retrieved from AWS Architecture Blog: <https://aws.amazon.com/blogs/architecture/lets-architect-serverless-architecture-on-aws/>
- Microsoft. (2023). *Multicloud solutions with the Serverless Framework*. Retrieved from Azure Architectures: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/serverless/serverless-multicloud>
- Microsoft. (2023). *Switch views during a video call*. Retrieved from Microsoft 365 Support: <https://support.microsoft.com/en-us/office/switch-views-during-a-video-call-3a4247bf-09ca-484a-a9f3-b96d1906a75e>
- Microsoft. (2023). *Why create typescript?* Retrieved from Typescript lang: <https://www.typescriptlang.org/why-create-typescript>
- Microsoft Azure. (2023). *Visual Studio Code*. Retrieved from Azure: <https://azure.microsoft.com/en-us/products/visual-studio-code/>

National Society of High School Scholars. (2020, Julio 28). *How to engage students in a virtual learning environment*. Retrieved from NSHSS Blog: <https://www.nshss.org/blog/how-to-engage-students-in-a-virtual-learning-environment/>

Nilson, L. B., & Goodson, L. A. (2017). *Online Teaching at Its Best*. Jossey-Bass.

Open AI. (2023). *About*. Retrieved from Open AI: <https://openai.com/about>

Open AI. (2023). *Chat - API Reference*. Retrieved from OpenAI API documentation: <https://platform.openai.com/docs/api-reference/chat>

OpenAI. (2023). *GPT-4 System Card*. OpenAI.

OpenAI. (2023). *GPT-4 Technical Report*. aiXiv.

OpenJS Foundation. (2023). *About Node.js*. Retrieved from Node.js: <https://nodejs.org/en/about>

Radford, A., Narasimhan, K., Salimans, T., & Ilya, S. (2019). *Improving Language Understanding by Generative Pre-Training*. OpenAI.

Red Hat. (2022, May 2011). *What is serverless?* Retrieved from Red Hat Topics: <https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless>

Rehkopf, M. (2023). *What is continuous integration?* Retrieved from Atlassian: <https://www.atlassian.com/continuous-delivery/continuous-integration>

Reselman, B. (2021, May 6). *The pros and cons of the Pub-Sub architecture pattern*. Retrieved from Red Hat: <https://www.redhat.com/architect/pub-sub-pros-and-cons>

Richardson, R., & Fröhner, D. (2019, Noviembre 22). *Application integration patterns for microservices: Fan-out strategies*. Retrieved from AWS Compute Blog: <https://aws.amazon.com/blogs/compute/application-integration-patterns-for-microservices-fan-out-strategies/>

- Rosenshine, B. (2012). Principles of Instruction, Research-Based Strategies That All Teachers Should Know. *American Educator*, 12-39.
- Ryan, M. R., & Deci, E. L. (2000). Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist* 55, 68-78.
- Sadalage, P., & Fowler, M. (2016, Mayo). *Evolutionary Database Design*. Retrieved from Martin fowler: <https://martinfowler.com/articles/evodb.html>
- Selvi, J. (2022, Diciembre 5). *Exploring Prompt Injection Attacks*. Retrieved from nccgroup: <https://research.nccgroup.com/2022/12/05/exploring-prompt-injection-attacks/>
- Sequelize. (2023, Julio 5). *Migrations*. Retrieved from Sequelize API Reference: <https://martinfowler.com/articles/evodb.html>
- Sequelize. (2023). *Sequelize-typescript*. Retrieved from Github: <https://github.com/sequelize/sequelize-typescript>
- Serverless Inc. (2023). *Serverless Infrastructure Providers*. Retrieved from Serverless Framework Documentation: <https://www.serverless.com/framework/docs/providers>
- Shawi, M. A. (2020, November 3). *Architecting for Reliable Scalability*. Retrieved from AWS Architecture Blog: <https://aws.amazon.com/blogs/architecture/architecting-for-reliable-scalability/>
- Solanki, J. (2021, Junio 1). *How to Avoid AWS Lambda Cold Starts?* Retrieved from SIMFORM Blog: <https://www.simform.com/blog/lambda-cold-starts/>
- Stack Overflow. (2023, Mayo). *2023 Developer Survey*. Retrieved from Stack Overflow Survey: <https://survey.stackoverflow.co/2023/>

- Subramainan, L., & Mahmoud, M. A. (2020). A Systematic Review on Students' Engagement in Classroom: Indicators, Challenges and Computational Techniques. *(IJACSA) International Journal of Advanced Computer Science and Applications* 11, 105-115.
- Symphony Solutions. (2020). *Why Cloud computing scalability matters for business growth*. Retrieved from Symphony Solutions Insights: <https://symphony-solutions.com/insights/cloud-computing-scalability>
- Tabassum, T., Allen, A. A., & De, P. (2020). Non-intrusive Identification of Student Attentiveness and Finding Their Correlation with Detectable Facial Emotions. *ACM Southeast Conference - ACMSE 2020 - Session 1*, 127-134.
- Terraza, D. F., Amaya, M., Piedrahita-Carvajal, A., Rodriguez-Marin, P. A., Duque-Muñoz, L., & Martinez-Vargas, J. D. (2022). Student's Attention Monitoring System in Learning Environments based on Artificial Intelligence. *IEEE Latin America Transactions*, 126-132.
- Trabelsi, Z., Alnajjar, F., Ambali Parambil, M., Gochoo, M., & Ali, L. (2023). Real-Time Attention Monitoring System for Classroom: A Deep Learning Approach for Student's Behavior Recognition. *Big Data And Cognitive Computing*, 1-17.
- Tsai, Y.-S. (2022). *Learning Analytics In a Nutshell*. Retrieved from SOLAR - Society for Learning Analytics Research: <https://www.solaresearch.org/about/what-is-learning-analytics/>
- Vernon, V. (2013). *Implementing Domain-Driven Design*. New Jersey: Addison-Wesley.
- Wang, S. (2021). Online Learning Behavior Analysis Based on Image Emotion Recognition. *International Information and Engineering Technology Association*, 865-873.

Willison, S. (2022, Septiembre 12). *Prompt injection attacks against GPT-3*. Retrieved from Simon Wilinon Weblog: <https://simonwillison.net/2022/Sep/12/prompt-injection/>

Yang, K., Wang, C., Sarsenbayeva, Z., Tag, B., Dingler, T., Wadley, G., & Goncalves, J. (2021). Benchmarking commercial emotion detection systems using realistic distortions of facial image datasets. *The Visual Computer*, 1447-1466.

9. ANEXOS

9.1. ANEXO 1 - REPOSITORIOS DEL CÓDIGO FUENTE DEL PROTOTIPO

<https://github.com/LucasVera/screenshot-analyzer>

<https://github.com/LucasVera/eafit-teacher-feedback-fe>

<https://github.com/LucasVera/eafit-teacher-feedback-be>

<https://github.com/LucasVera/eafit-learning-analytics-be>

<https://github.com/LucasVera/eafit-screenshot-app>