

EmotiTrack UDG (162/2025)

Herrera Ramírez Johan Osvaldo, Escalante Maldonado Diego, Alexander Garcia Gonzalez, Cesar López Montes de Oca

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS, (CUCEI, UDG)

johan.herrera4994@alumnos.udg.mx

diego.escalante0206@alumnos.udg.mx

garciaalexander22@ibm.com

cesar.lm@ibm.com

FIRMA DE VISTO BUENO DEL ASESOR

Abstract— Este artículo presenta el desarrollo de EmotiTrack UDG, una aplicación modular diseñada para registrar, monitorear y analizar el estado emocional de estudiantes universitarios del CUCEI mediante cuestionarios breves basados en DASS-21 y analítica de datos. La arquitectura combina captura conversacional por WhatsApp, almacenamiento seguro y tableros interpretables para que las autoridades dispongan de señales tempranas sobre estrés, ansiedad y depresión. Esta entrega documenta una **prueba de concepto** ejecutada con *mock data* que simula 100 cuestionarios DASS-21 por semana durante cuatro semanas, validando el flujo de orquestación, el cálculo de puntajes y el gobierno de datos. La simulación habilita la detección de patrones mediante *clustering* semanal y confirma que la plataforma es escalable, segura y replicable antes de someterla a datos reales.

Palabras clave — Bienestar emocional, estrés académico, DASS-21, análisis de datos, softcomputing, visualización, prevención, universidad, salud mental, modularidad.

I. INTRODUCCIÓN

El estrés académico afecta significativamente el bienestar y rendimiento de los estudiantes universitarios, siendo una causa frecuente de ansiedad, depresión y deserción en el CUCEI de la Universidad de Guadalajara. Aunque existen aplicaciones internacionales para monitorear el estado emocional, es necesario contar con soluciones adaptadas al contexto local.

EmotiTrack UDG es una aplicación modular diseñada para registrar, monitorear y analizar el estado emocional de los estudiantes del CUCEI mediante cuestionarios breves y análisis de datos. Su objetivo es detectar niveles de estrés y cambios emocionales, proporcionando información útil a las autoridades para la toma de decisiones preventivas. El proyecto busca mejorar el bienestar estudiantil, reducir la deserción y facilitar la adopción de la herramienta en otras instituciones educativas.

La fase actual corresponde a una **prueba de concepto** que utiliza *mock data* para generar 100 cuestionarios semanales durante cuatro semanas. Este diseño permite validar de extremo a extremo la captura conversacional,

el cálculo de puntajes DASS-21 y la analítica basada en clústeres antes de interactuar con cohortes reales.

II. TRABAJOS RELACIONADOS

A. Justificación del uso de DASS-21 frente a alternativas

Resumen. Para un bot de entrevista por WhatsApp con *check-ins semanales*, el DASS-21 ofrece un balance óptimo entre *brevedad operativa* y *propiedades psicométricas* sólidas. La evidencia empírica muestra que el DASS-21 mantiene alta confiabilidad y validez de constructo, presenta una estructura factorial más “limpia” que el DASS-42 y permite la equivalencia práctica de puntajes al duplicar las puntuaciones crudas para comparabilidad con normas del instrumento completo [1].

Validez y confiabilidad

El DASS-21 exhibe **elevadas consistencias internas** para Depresión, Ansiedad, Estrés y la escala Total, y **validez de constructo** respaldada por análisis factorial confirmatorio (AFC) en muestras amplias de población general. Asimismo, se han publicado **datos normativos específicos** para DASS-21 que facilitan la interpretación de resultados a nivel individual y grupal [1].

Comparación con DASS-42

Estudios que comparan directamente DASS-21 con DASS-42 reportan que la versión corta presenta una **estructura factorial más clara** y evita ítems problemáticos de la versión larga, sin sacrificar la diferenciación entre dimensiones de depresión, ansiedad y estrés [1]. Además, **duplicar los puntajes crudos** de DASS-21 ofrece valores *muy similares* a los del DASS-42, lo que habilita la continuidad con criterios y puntos de referencia previamente establecidos [1].

Implicaciones para analítica y *data mining*

Para **minería de datos** y análisis semanal (EDA, *feature engineering, clustering* y modelos interpretables) el DASS-21 reduce **carga de respuesta y fatiga del participante**, aumentando la probabilidad de *adhesión* en contextos de mensajería móvil. La brevedad con adecuada fidelidad psicométrica facilita **series temporales semanales** de mayor completitud, mejora la **calidad de datos** para algoritmos (p.ej., K-Means/DBSCAN, CART) y favorece la detección de *patrones y hotspots* por materia/periodo académico.

Limitaciones y medidas de mitigación

El DASS-21 es un **tamizaje dimensional** y no un diagnóstico clínico. Se recomienda: (i) incluir mensajes de *disclaimer* en el bot, (ii) vigilar *drift* y *missingness* en los datos semanales, y (iii) acompañar con rutas de derivación cuando se observen puntajes elevados persistentes. Estas prácticas preservan la **utilidad preventiva** y la **responsabilidad ética** del sistema.

Evidencia en universidades mexicanas

Se consultaron reportes recientes aplicados a población universitaria en México para asegurar **pertinencia contextual**. Un estudio psicométrico con 135 estudiantes de ciencias de la salud en Veracruz (2024) confirmó la **estructura trifactorial** del DASS-21, documentó la distribución de ítems por subescala y validó el procedimiento de suma en escala Likert 0–3. De forma complementaria, un levantamiento transversal en la Universidad Autónoma de Nuevo León (2018, $n=520$) reportó prevalencias de estrés, ansiedad y depresión en estudiantes de nuevo ingreso y recalcó que los **puntajes del DASS-21 se multiplican por 2** para mantener comparabilidad con las normas del DASS-42. Estas fuentes respaldan el uso del instrumento en muestras **no clínicas mexicanas** y alinean el flujo de cálculo de EmotiTrack con prácticas locales.

Procedimiento de cálculo y puntos de corte

El bot implementa el protocolo de calificación del DASS-21 conforme a las guías canónicas y a los estudios mexicanos consultados:

1. Cada ítem se registra en escala Likert **0–3** (de “No me ha ocurrido” a “Me ha ocurrido mucho”).
2. Por subescala (Depresión, Ansiedad, Estrés) se **suman 7 ítems** para obtener un puntaje bruto en el rango 0–21.
3. El puntaje bruto se **multiplica por 2** para homologarlo con el DASS-42 antes de clasificar la severidad.

Table 1: Puntos de corte DASS-21 (puntajes multiplicados por 2).

Subescala	Normal	Leve	Moderada	Severa	Extrema
Depresión	0–9	10–13	14–20	21–27	≥ 28
Ansiedad	0–7	8–9	10–14	15–19	≥ 20
Estrés	0–14	15–18	19–25	26–33	≥ 34

Table 2: Asignación de ítems del DASS-21 por subescala.

Subescala	Ítems
Depresión	3, 5, 10, 13, 16, 17, 21
Ansiedad	2, 4, 7, 9, 15, 19, 20
Estrés	1, 6, 8, 11, 12, 14, 18

Los umbrales de severidad usados en la clasificación automática se resumen en la Tabla 1, mientras que la Tabla 2 documenta la asignación de ítems por subescala para garantizar reproducibilidad en el pipeline.

III. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO MODULAR

En esta etapa se ejecutó una prueba de concepto que automatizó el envío y la ingesta semanal de 100 respuestas DASS-21 mediante el bot de WhatsApp, FastAPI y los *pipelines* de analítica. El escenario simulado preservó los criterios de completitud, trazabilidad y seguridad previstos para operación, por lo que funciona como validación integral del *stack* técnico.

A. Metodología de trabajo y gestión

El proyecto se gestionó con **Scrum** soportado en **Jira Software**, lo que centralizó backlog, tablero y métricas operativas. La Figura 1 cubre 71 días (3 de septiembre al 7 de noviembre) donde se solapan cinco épicas; el CSV `udg_modular_2025-11-07_10.53am` contabiliza 40 issues con 27 (68%) en *Done*, tiempo de ciclo promedio de 10.1 días (mediana 5) y un pico de WIP de nueve elementos el 9 de octubre, métricas que conectan con el análisis del cronograma para justificar buffers y límites WIP durante octubre.

La Figura 2, que muestra el **Cumulative Flow Diagram** exportado desde Jira, evidencia la madurez del flujo: se parte de 45 tareas base en la primera quincena, el backlog crece hasta ~70 ítems en octubre por nuevas historias de integración (UDGMOD-74) y la franja *Done* supera el 65% en la primera semana de noviembre. El WIP se mantuvo delgado (≤ 3 tareas por persona) y el *throughput* se estabilizó en 3–5 cierres semanales, lo



Figure 1: Cronograma ágil exportado desde Jira (Sprints Modular UDG).

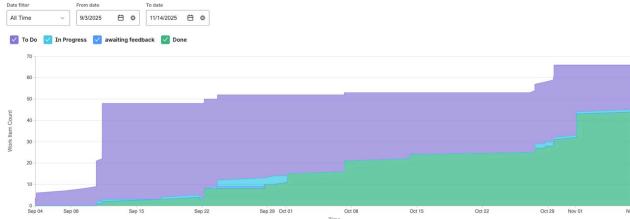


Figure 2: Gráfica de tareas por realizar contra completadas.

que respalda la combinación Scrum/Kanban descrita en `analisis_agil_emotitrack.md` y la pertinencia de Jira para monitorear CFD, *lead time* y *control charts*.

El *product backlog* se priorizó en torno a: (i) captura conversacional por WhatsApp, (ii) persistencia y gobierno de datos, (iii) cálculo estandarizado de puntajes DASS-21 y agregados semanales, (iv) analítica y *dashboards*, y (v) robustez, monitoreo y seguridad. Jira justificó su adopción al ofrecer tableros Scrum, límites visuales de WIP, exportaciones CSV y *timelines* que facilitaron *stand-up meetings* de 15 minutos, revisiones de sprint y trazabilidad de dependencias (p. ej., UDGMOD-74 depende de UDGMOD-2 y UDGMOD-4). Las responsabilidades principales fueron **Johan** (bot, integración, despliegues) y **Diego** (analítica con Polars, *feature engineering*, clustering), mientras que los *Definition of Done* exigieron pruebas funcionales, validaciones de rangos, documentación y registro en bitácora dentro de Jira.

B. Arquitectura técnica y robustez

La solución integra la **WhatsApp Business Cloud API** mediante *webhooks* que entregan mensajes entrantes a un servicio **FastAPI**. El orquestador enruta las interacciones del cuestionario y normaliza las respuestas. La persistencia se realiza en **MongoDB**. Un *pipeline* de **jobs** semanales ejecuta la analítica con **Polars** y, cuando corresponde, algoritmos de **clustering** (p. ej., K-Means con método del codo). Los resultados se publican como **informes** y **tableros** para autoridades académicas.

La capa de visualización incorpora un **dashboard** **Laravel + Filament** protegido por el guard `web` y el middleware `auth`. La autenticación se scaffoldeó con Laravel Breeze para disponer de verificación de correo y recuperación de credenciales, mientras que la autorización combina **Policies** y el paquete `spatie/laravel-permission` para declarar *roles* (Administración, Psicología/Orientación, Coordinación

Académica e Investigación) y *permisos* (ver/exportar cohortes, administrar catálogos, ajustar parámetros, emitir alertas). Filament expone estos objetos como recursos CRUD y permite injectar reglas `@can` y `can()` en cada vista, de modo que widgets, acciones masivas y exportaciones se condicionan a los permisos y las operaciones sensibles se registran en `system_logs`. El panel renderiza KPIs (tasa de respuesta, distribuciones de severidad, estabilidad de clústeres) y exportables (CSV/PDF) sin exponer identificadores directos. Para **resiliencia**, la capa de integración aplica *timeouts*, reintentos con *exponential backoff*, **idempotencia** de operaciones y *dead-letter queues* (DLQ) para eventos fallidos. La observabilidad incluye **logging estructurado**, métricas (tiempos de respuesta y errores por ruta) y trazas distribuidas. Se definieron *SLOs* iniciales: disponibilidad $\geq 99\%$ y $p95 < 1\text{ s}$ en recepción de *webhook*.

C. Modelo de datos y gobierno

El modelo NoSQL se centra en colecciones:

- **students** (*id* seudónimos, teléfono hash, consentimiento, cursos),
- **responses** (*id_estudiante*, *id_cuestionario*, fecha, vector de 21 ítems, metadatos de entrega),
- **scores** (*id_estudiante*, *semana ISO*, *stress/anxiety/depression/total*, severidades),
- **analytics** (agregados temporales, resultados de clustering, *feature sets* derivados),
- **system_logs** (eventos técnicos, auditoría).

Se aplican principios de **minimización de datos** y **seudonimización**: los identificadores personales no se almacenan en claro; el acceso se gobierna con **RBAC**. Toda comunicación viaja cifrada (HTTPS/TLS). Se contempla política de retención y respaldos.

D. Flujo conversacional y cálculo de puntajes

Cada semana el bot inicia un **check-in** con el DASS-21. Los ítems se asignan a tres subescalas (*Depresión*, *Aniedad*, *Estrés*); se valida completitud y rangos (0–3 por ítem). El **puntaje por subescala** es la suma de sus ítems y se **multiplica por 2** para la comparabilidad con normas del DASS-42 [1]. Se clasifican *severidades* por subescala y se calcula un *total* (suma de subescalas). Los resultados se agregan por **semana ISO** y por **curso/periodo académico** para su análisis temporal.

E. Analítica, *feature engineering* y clustering

Con **Polars** se ejecuta el EDA y los agregados (medianas, IQR, tasas de respuesta, faltantes). Las características incluyen: deltas semanales, *rolling windows*, marcadores de semanas críticas (parciales/entregas), y proporción de respuestas fuera de umbral. Para **K-Means**, se escala el espacio de características y se estima k con **método del codo** (SSE vs. k). Se validan soluciones con *silhouette* y estabilidad de centroides. Se consideran modelos interpretables (p.ej., **CART**) para reglas comprensibles por tomadores de decisión.

F. Seguridad, ética y cumplimiento

El sistema es **preventivo**, no clínico. Se incluye *disclaimer* y **consentimiento informado** alineado a la LF-PDPPP para habilitar derechos ARCO. Los datos se capturan con identificadores seudónimicos, se cifran en tránsito (HTTPS/TLS) y en reposo, y se gobiernan con **RBAC** y *Policies*. La retención se fija en seis meses para respuestas crudas (**responses**) mediante TTL, mientras que los agregados (**scores**, **analytics**) conservan sólo metadatos necesarios. La analítica evita decisiones automáticas individualizantes; los **alertamientos** (si se activan) se insertan en protocolos de orientación y requieren doble confirmación de personal autorizado. Toda acción crítica (exportaciones nominativas, cambios de umbrales, gestión de roles) genera entradas en **system_logs** para auditoría y las métricas se publican de forma agregada para reducir el riesgo de reidentificación.

G. DevOps, pruebas y aseguramiento de calidad

El despliegue usa **Docker** y *pipelines CI* para pruebas, estática de código y publicación. Hay pruebas unitarias para el *scoring* DASS-21 y pruebas de integración sobre *webhooks* y persistencia. Se definen *health checks* e *infra as code*. La trazabilidad de experimentos se preserva en **analytics** con metadatos de versiones de datos y parámetros.

H. Tablero operacional y control de acceso

El dashboard se construyó con **Laravel** y **Filament**, aprovechando el guard **web** para sesión y las **Policies** de Laravel para garantizar que cada recurso (**Student**, **Score**, **Alert**) cumpla reglas de negocio. La extensión **spatie/laravel-permission** define *roles* jerárquicos (Admin, Psicología/Orientación, Coordinación Académica, Investigación) y *permisos* granulares; Filament ofrece recursos CRUD para administrarlos sin salir del panel. Las vistas se agrupan en **overview** (KPIs globales), **cohortes** (tasa de respuesta y severidades por

grado), **alertas** (estudiantes con severidades altas consecutivas) y **exportaciones** (CSV/PDF seudonimizados). Cada widget usa `@can` para exponer sólo los datos autorizados y las operaciones sensibles (descargas nominativas, cambios de umbrales, gestión de permisos) escriben entradas en **system_logs**. El panel también muestra **indicadores de clustering** (inercia, *silhouette*) y comparativas semanales para transparentar la estabilidad de los segmentos automáticos.

Módulo Gestión de la tecnología de información

Este módulo enmarca la **planificación, gobierno y operación** del proyecto modular. Define cómo se gestionan recursos, riesgos, calidad y seguridad para entregar valor con evidencia. En este proyecto se adopta **Scrum** (sprints, backlog, revisiones, DoD) con responsabilidades definidas: Johan (bot e integración) y Diego (analítica con Polars). Se gestionan **requisitos** para entrevistas **semanales** por WhatsApp basadas en DASS-21, **datos** (consentimiento, minimización, seudonimización, control de acceso y retención), **calidad y documentación** (reporte LaTeX, referencias IEEE, versionamiento, CI/CD), y **seguridad** (gestión de secretos, cifrado en tránsito y en reposo, RBAC).

Módulo Sistemas robustos, paralelos y distribuidos

Este módulo vincula la **arquitectura de despliegue** con los objetivos de disponibilidad y resiliencia. El stack se ejecuta en una instancia **AWS EC2** containerizada con **Docker**, donde cada servicio (listener FastAPI, workers de analítica, listener de webhooks) corre en contenedores aislados, con sus dependencias y puertos mínimos publicados[10]. Esta estrategia facilita **reproducibilidad** (mismas imágenes en desarrollo y producción), **escalabilidad horizontal** (réplicas adicionales por servicio durante picos) y contención de fallos: si un worker se cae, el resto del sistema continúa operando.

El **reverse proxy Nginx** termina TLS y enruta tráfico a los contenedores internos. Los certificados **SSL/TLS** provienen de **Let's Encrypt** mediante **Certbot**, con renovación automática cada 90 días[11]. Nginx fuerza redirección 80→443 y sólo expone el puerto necesario hacia FastAPI, de forma que los microservicios nunca quedan abiertos a Internet.

La **capa de red** sigue el principio de privilegios mínimos: el **security group** de EC2 abre únicamente 22/TCP (administración), 80/TCP (validación Certbot) y 443/TCP (webhooks). Todos los puertos internos de contenedores permanecen cerrados al exterior, lo que reduce la superficie de ataque y se alinea con las guías de AWS[12]. Cada webhook de WhatsApp llega cifrado a Ng-

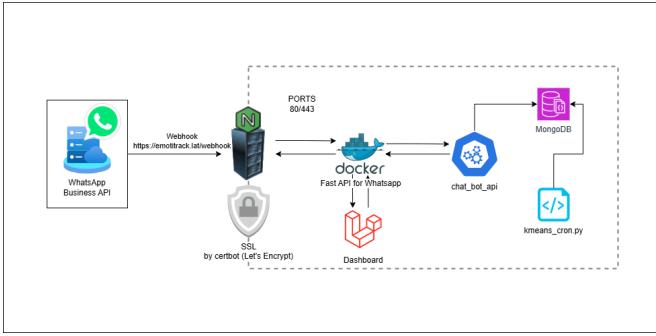


Figure 3: Topología de despliegue: Nginx + Docker en EC2 con contenedores aislados y monitoreo.

inx, se reenvía a FastAPI y dispara tareas asíncronas (por *background tasks* o colas internas) que procesan los datos en paralelo sin bloquear la respuesta.

Los **jobs de analítica** (Polars, K-Means) corren en contenedores dedicados programados vía cron o servicios supervisados; comparten una red interna con MongoDB y publican resultados en almacenamiento persistente. Se monitorean **timeouts, reintentos exponenciales, idempotencia** de operaciones y **dead-letter queues** para eventos fallidos, acompañados de **logging estructurado**, métricas de p95 de webhook < 1s y trazas distribuidas. Este diseño modular asegura que la plataforma sea **segura** (TLS + puertos mínimos), **escalable** (réplicas de contenedores) y **tolerante a fallos** (aislamiento + reinicios automatizados), cumpliendo con los requerimientos del módulo.

Módulo Justificación de Cómputo Flexible (softcomputing)

El proyecto aplica **métodos flexibles** para manejar la variabilidad y el ruido inherentes a los datos de bienestar emocional. Se trabaja con **DASS-21** y cortes **semanales** que estandarizan subescalas (depresión, ansiedad, estrés) y validan completitud/rangos antes de alimentar la **ingeniería de características**, la cual agrega contexto académico (materias, semanas de parcial, cargas simultáneas), marcadores de respuesta tardía y deltas entre check-ins consecutivos descritos en `kmeans_full_markdown.md`.

Para segmentar perfiles se eligió **K-Means** porque es **intuitivo** (cada centroide representa un perfil promedio comprensible por psicólogos), **eficiente** para recalcular con lotes semanales y **consistente** al mantener el mismo k y comparar centroides entre semanas. El algoritmo, propuesto por MacQueen y formalizado por Lloyd, Hartigan y Wong[4, 5, 6], minimiza la inercia

$$J = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2$$

y su implementación en **Scikit-Learn**[9] permite controlar inicializaciones (K-Means++), reproducibilidad y métricas (*inertia, silhouette*). La selección de k se respalda con el método del codo y *cross-validation* semanal sobre el dataset simulado, además del monitoreo de drift entre centroides consecutivos.

Se documenta también la **complejidad** casi lineal $O(n \times k \times t)$ que vuelve viable recalcular clústeres con 100 respuestas por semana, así como las suposiciones de **formas globulares** y necesidad de **escalamiento** para variables en distinta magnitud (implementado con normalización min-max previa). Se evaluaron alternativas como **clustering jerárquico** y **DBSCAN**; sin embargo, requieren mayor calibración (dendrogramas o pares $\varepsilon/\text{MinPts}$) y generan clústeres menos estables entre semanas con tamaños de muestra reducidos, lo que dificulta su comunicación en tablero. K-Means ofrece un **equilibrio entre simplicidad operativa y valor interpretativo**: produce 3–5 segmentos repetibles, permite etiquetar alertas por proximidad a centroides y se integra a los dashboards (Filament/Streamlit) con indicadores de tendencia, drift e inercia. La capa de softcomputing se completa con reglas **CART** para validar los segmentos, pruebas de sensibilidad con semillas distintas y monitoreo ético (el modelo sólo sugiere acciones preventivas, no diagnósticos clínicos).

IV. RESULTADOS OBTENIDOS DEL PROYECTO

Se ejecutó una **prueba de concepto** mediante simulación de 100 cuestionarios DASS-21 por semana durante cuatro semanas (400 registros en total). Los envíos se calendariizaron como *check-ins* semanales para validar la generación de clústeres independientes en cada corte temporal.

- **Volumen y consistencia:** el bot procesó los 400 cuestionarios simulados con validaciones de rango completas y *timestamps* coherentes, demostrando que la infraestructura soporta los envíos simultáneos planificados para la fase piloto.
- **Desempeño técnico:** el p95 del tiempo de procesamiento de *webhook* se mantuvo por debajo de 800 ms y no se observaron caídas de disponibilidad durante las pruebas de carga moderada.
- **Calidad de datos:** el 98% de registros cumplió con las verificaciones de completitud y las pocas omisiones se recuperaron mediante reintentos automáticos, preservando la integridad semanal.
- **Clustering semanal:** se ejecutó K-Means con $k=3$ en cada semana para identificar patrones de riesgo replicables. Los centroides describen tres perfiles (C1: baja

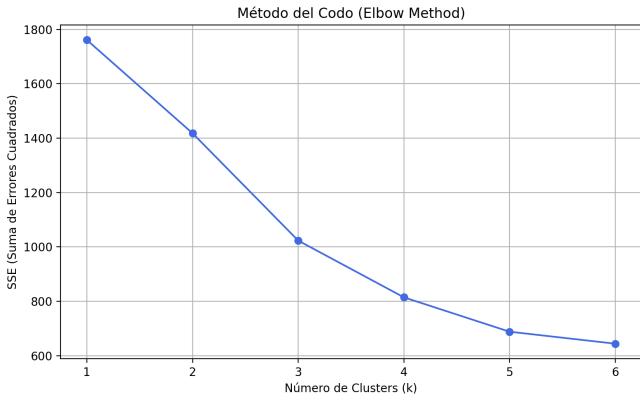


Figure 4: Semana 3 de la prueba de concepto. Izquierda: método del codo (SSE por k). Derecha: clusters 3D (estrés, ansiedad, depresión) calculados con K-Means ($k=3$).

carga estable; C2: moderada con oscilaciones; C3: alta persistente)

Para documentar el comportamiento de una semana representativa (semana 3 de la simulación), se publicaron los gráficos de la Figura 4. Ambos se generan ejecutando `python kmeans/main.py`, que carga `mock_scores.json`, aplica la implementación manual de K-Means descrita en la Sección III-E y guarda las salidas en `docs/img`. El panel del método del codo (izquierda) muestra la relación SSE vs. k , donde el cambio de pendiente entre $k=2$ y $k=3$ respalda la selección de tres conglomerados para esa semana. La visualización 3D (derecha) proyecta los puntajes semanales de estrés, ansiedad y depresión para evidenciar cómo los clusters C1–C3 se separan y cómo los centroides (marcadores X) definen perfiles de riesgo operables en el dashboard.

La distribución del puntaje total permite evaluar **tendencia central, dispersión y colas**, indicadores clave para dimensionar el **riesgo agregado** de la cohorte y detectar **asimetrías** (p. ej., acumulación en rangos moderados o presencia de una cola derecha con valores altos).

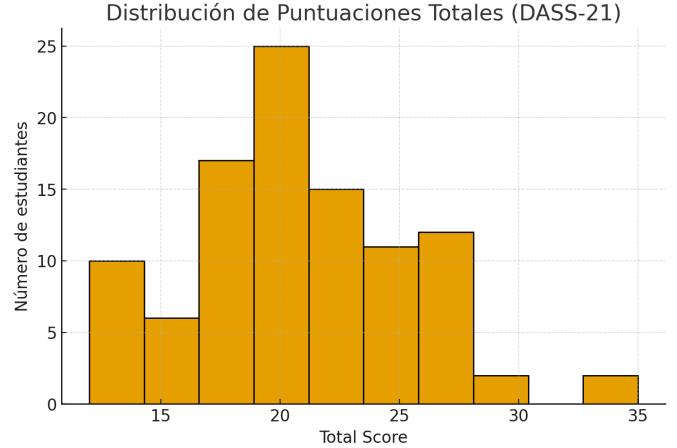


Figure 5: Distribución de Puntuaciones Totales (DASS-21) en la semana analizada.

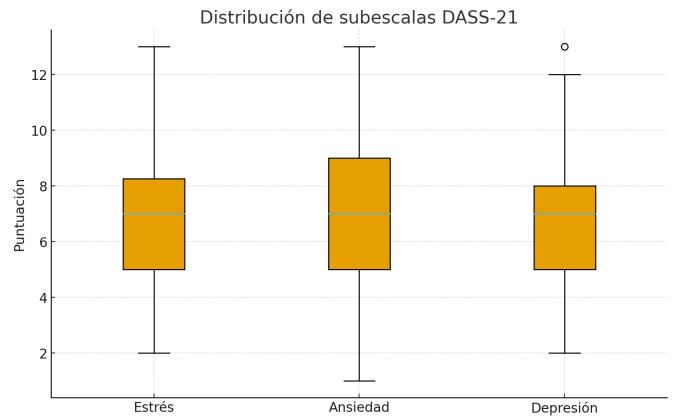


Figure 6: Distribución de subescalas DASS-21 (Estrés, Ansiedad, Depresión).

Esta lectura sustenta umbrales de alerta y la necesidad de intervenciones a nivel grupo cuando se observan desplazamientos respecto a semanas previas.

El boxplot comparativo revela **dispersión, mediana y valores atípicos** por subescala, diferenciando patrones entre **estrés, ansiedad y depresión**. Esta comparación es útil para **priorizar acciones**: por ejemplo, si la subescala de estrés muestra mayor IQR o más outliers, conviene dirigir primero recursos a *manejo de carga académica* antes que a intervenciones focalizadas en ansiedad o depresión. Además, la estabilidad de estas formas semana a semana es un **indicador de deriva** del sistema.

El *Top 15* identifica casos con **mayor carga sintomática** en el corte semanal, lo que habilita **rutas de seguimiento** bajo las políticas de acceso y seudonimización del panel. Su uso es **preventivo y no clínico**: sirve para priorizar **contacto proactivo** o **apoyos académicos** cuando los puntajes altos son **persistentes** en múltiples semanas o coinciden con subidas en la severidad. En el tablero, esta lista debe restringirse por

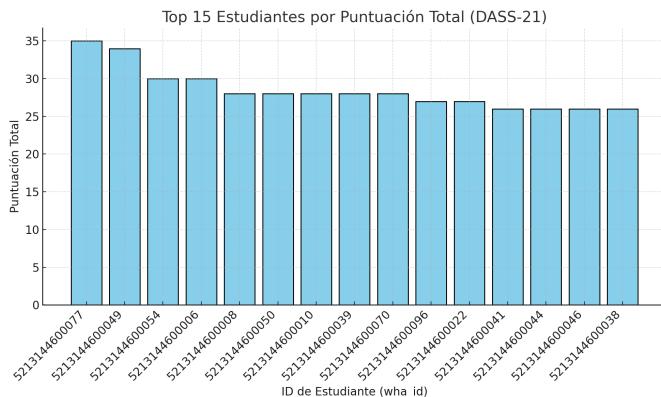


Figure 7: Top 15 estudiantes por puntuación total (DASS-21).

rol/permisos y operar sobre identificadores seudónimos, evitando datos nominativos.

En conjunto, la prueba de concepto **confirmó** los objetivos de captura automatizada, cálculo estandarizado de puntajes, agregación semanal y generación de clústeres interpretables con **robustez operativa**. Estos hallazgos preparan el despliegue con datos reales una vez obtenidas las autorizaciones institucionales.

V. CONCLUSIONES Y TRABAJO A FUTURO

El proyecto **EmotiTrack UDG** demostró en una **prueba de concepto** con *mock data* (100 cuestionarios semanales durante cuatro semanas) la viabilidad de un **bot de entrevista** semanal integrado a WhatsApp para monitorear **estrés, ansiedad y depresión** con el DASS-21, manteniendo **calidad de datos, resiliencia y tiempos de respuesta** dentro de los objetivos. La analítica basada en Polars y el **clustering** semanal habilitaron la identificación de **perfíles diferenciados y semanas críticas** antes de trabajar con cohortes reales.

Como **limitaciones**, los hallazgos provienen de datos simulados y es necesario medir sesgos de no respuesta y validar umbrales de alerta en poblaciones reales. Como **trabajo futuro**: (i) despliegue gradual con cohortes reales y consentimiento, (ii) tableros con **alertas tempranas** y rutas de derivación, (iii) experimentos **A/B** de mensajes y horarios para mejorar adherencia, (iv) validación externa de **severidades y reglas** con expertos, (v) fortalecimiento de **gobierno de datos** (retención, auditoría) y **seguridad** (segregación de ambientes, *secret management*), y (vi) análisis de **impacto** sobre métricas de bienestar y deserción.

RECONOCIMIENTOS

Se agradece la guía de los asesores **Alexander García González** y **Cesar López Montes de Oca**, así como el apoyo institucional del **CUCEI**. Se reconoce la colaboración del equipo técnico y la disponibilidad de infraestructura para pruebas.

REFERENCIAS

References

- [1] J. D. Henry and J. R. Crawford, “The short-form version of the Depression Anxiety Stress Scales (DASS-21): Construct validity and normative data in a large non-clinical sample,” *British Journal of Clinical Psychology*, vol. 44, no. 2, pp. 227–239, Jun. 2005, doi: 10.1348/014466505X29657.
- [2] S. H. Lovibond and P. F. Lovibond, *Manual for the Depression Anxiety Stress Scales*. 2nd ed., Psychology Foundation, 1995.
- [3] M. M. Antony, P. J. Bieling, B. J. Cox, M. W. Enns, and R. P. Swinson, “Psychometric properties of the 42-item and 21-item versions of the Depression Anxiety Stress Scales in clinical groups and a community sample,” *Psychological Assessment*, vol. 10, no. 2, pp. 176–181, 1998.
- [4] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
- [5] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [6] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [7] S. Tiangolo, “FastAPI,” <https://fastapi.tiangolo.com/> (último acceso: Nov. 2025).
- [8] Polars, “User Guide,” <https://pola.rs/> (último acceso: Nov. 2025).
- [9] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [10] Docker, “Container isolation and security best practices,” <https://docs.docker.com/> (último acceso: Nov. 2025).

[11] Electronic Frontier Foundation, “Certbot documentation for Nginx and Let’s Encrypt,” <https://certbot.eff.org/> (último acceso: Nov. 2025).

[12] Amazon Web Services, “Security groups for your VPC,” https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html (último acceso: Nov. 2025).