



*"Ayuda a las personas a encontrar
una nueva perspectiva en su vida y a
superar los desafíos."*



Enzo Camizan - enzo.camizan@utec.edu.pe

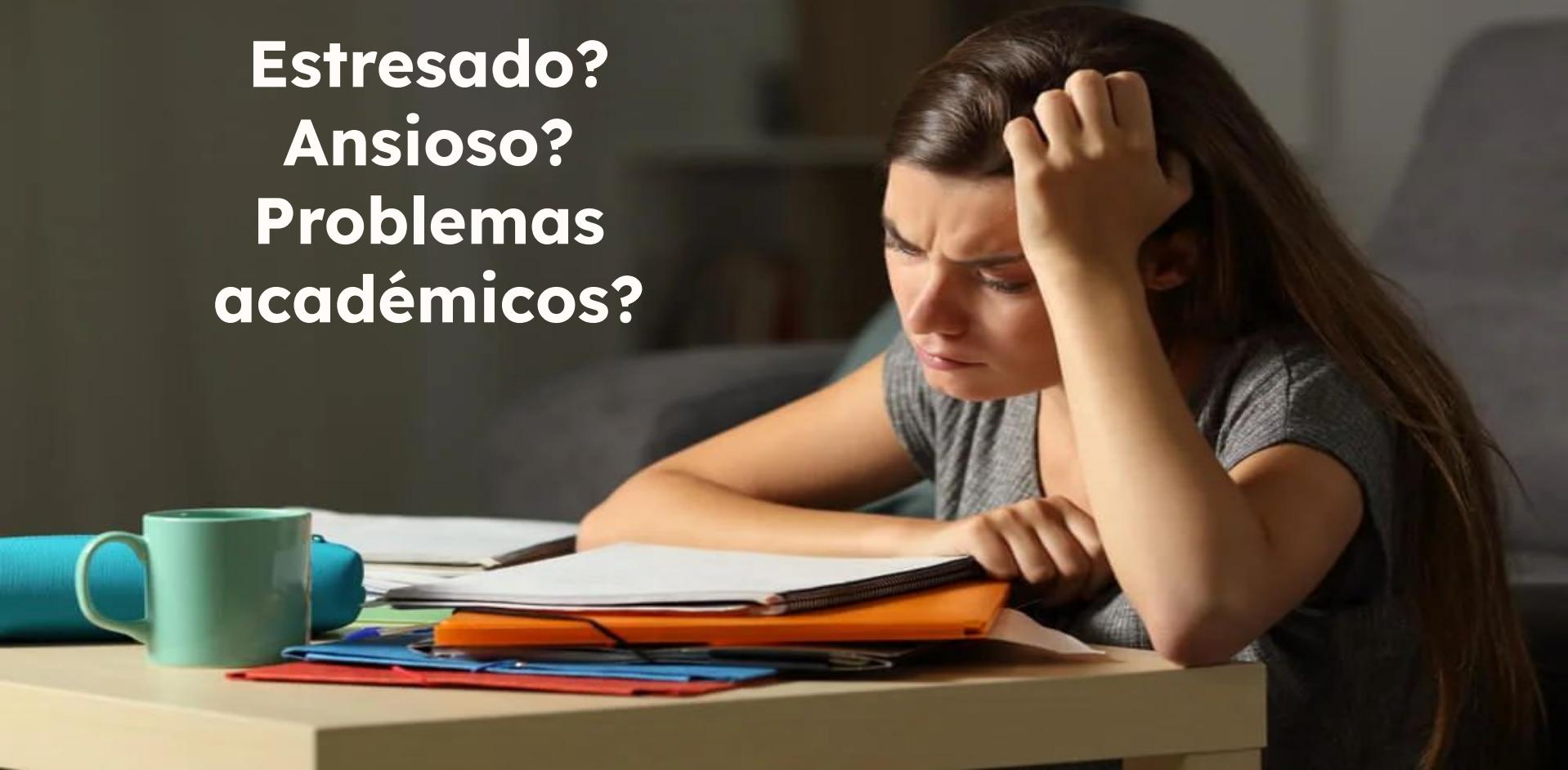
Paolo Vásquez - paolo.vasquez@utec.edu.pe

Valeria Espinoza - valeria.espinoza@utec.edu.pe

Valentín Quezada - hector.quezada@utec.edu.pe

Sofía García - sofia.garcia@utec.edu.pe

**Estresado?
Ansioso?
Problemas
académicos?**



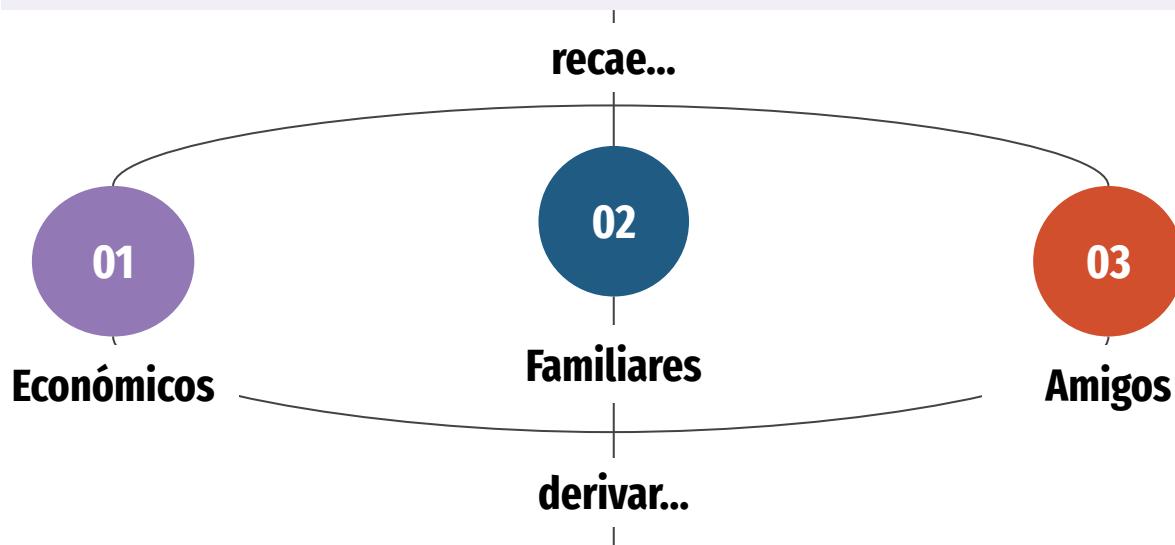
Contexto del problema



Bienestar y Estudios

)
afecta...

Salud Mental --- Universitarios



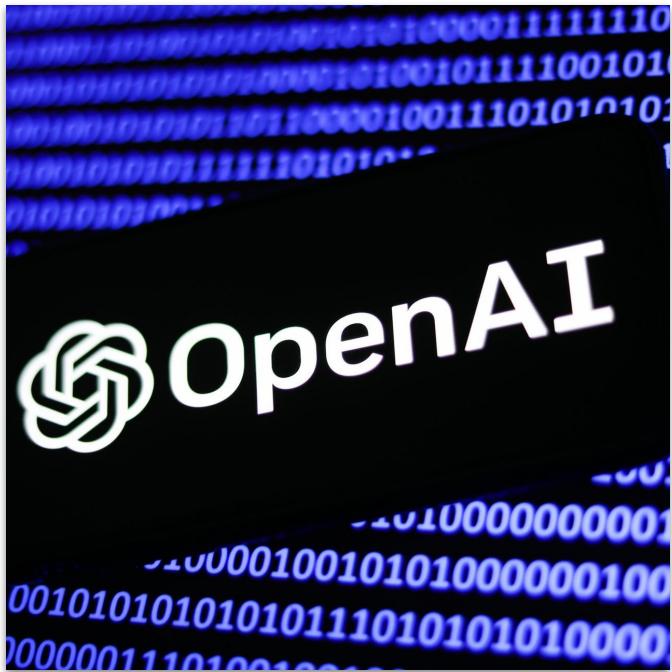
Ansiedad - Acoso - Discriminación - Distimia - Depresión



Dificultad para acceder a la ayuda necesaria

¿Qué es Aurora?

Situación actual





Solución

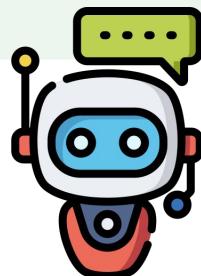
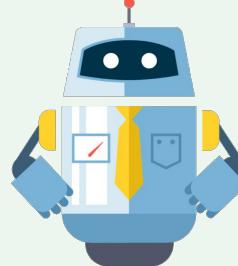
Aurora



Una solución que brinda esperanza

ChatBot enfocado en salud mental

Priorizado para Bienestar Estudiantil



Ayuda

Con una necesidad significativa actualmente



Facilita

La comunicación y búsqueda de consejos



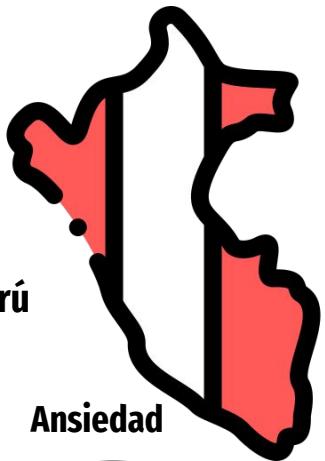
Previene

Situaciones problemáticas y perjudicantes



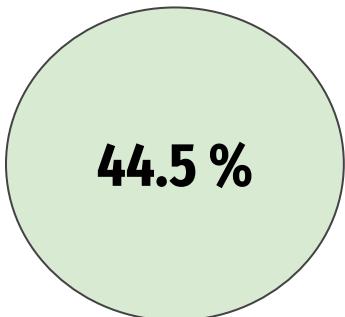


Depresión y ansiedad en el mercado

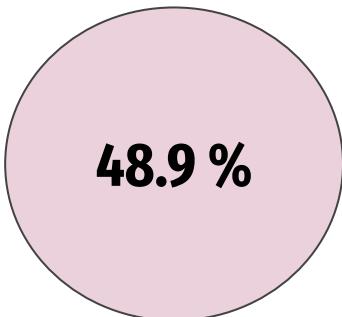


Prevalencia en LATAM

Depresión



Ansiedad

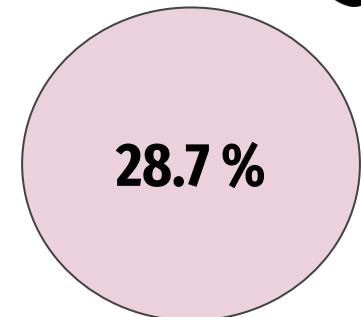


Prevalencia en Perú

Depresión



Ansiedad



Vicente, B., Kohn, R., Rioseco, P., Saldivia, S., Torres, S., & Levav, I. (2012). Lifetime and 12-month prevalence of DSM-III-R disorders in the Chile psychiatric prevalence study. *American Journal of Psychiatry*, 149(11), 1544-1552. doi: 10.1176/appi.ajp.149.11.1544

Cifras de suicidio en estudiantes universitarios de Lima, Perú

71.5%

de los intentos de suicidio en el Perú son realizados por personas de entre 15 y 34 años.

90%

de suicidas tienen problemas mentales que no han sido diagnosticados oportunamente.

(Perales et. al, 2019)

Causas de ideación suicida:

- **Estrés académico (52%)**
- Baja autoestima (50%)
- Problemas con padres (39%)
- Agresión sexual (32%)
- Problemas con pareja (20%)
- Problemas económicos (18%)

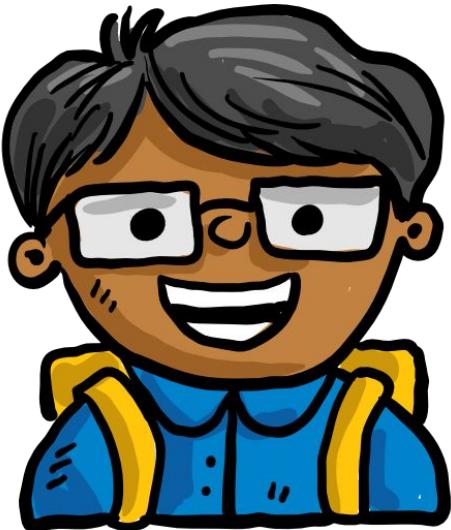
Consecuencias de ideación suicida:

- Cortarse las venas (29%)
- Tomar pastillas (29%)
- Ahorcamientos (21%)

(Perales et. al, 2019)



Usuario Modelo



- Paolo es un joven universitario de 19 años que está atravesando un periodo de inestabilidad emocional.
- Sus dificultades se centran en el hogar, la universidad y aspectos adicionales como, por ejemplo, una complicada relación con sus padres, o la falta de dinero y comida.
- Por otra parte, en la universidad sufre de burlas constantes, pues sus compañeros de clase le hacen comentarios negativos muy seguido y eso lo hace sentirse bastante criticado.
- Para buscar ayuda, a Paolo le recomiendan asistir al counter de bienestar estudiantil en el campus universitario. No obstante, es tímido y le cuesta abrirse con las personas tan fácilmente (aún por correo).
- Paolo descubre Aurora, una consejera universitaria virtual.

Modelo de conversación (script)

- “Hoy como que no me he sentido bien conmigo mismo”
- Es que siento que no puedo hablar con nadie y pucha, no me gusta contarlo
- Siento que la carga académica es fuerte y no ir bien en los estudios me está haciendo sentir frustrado 24/7
- Hay algo que podría hacer para sentirme mejor? Sin tener que hablarle a nadie sobre esto
- Y hay alguna serie o canción que recomendarías, para no sentirme tan equis???
- No sé, si siento que quiero morir, hay alguien a quien le podría hablar?



¿Qué resultado se espera?

- Sentimiento de alivio al poder expresar tus problemas y ser escuchado/a por el chatbot.
- Identificación de patrones de pensamiento negativo o destructivo relacionados con tu relación con tus padres, la falta de dinero y comida, o tus estudios
- Aumento de la empatía hacia ti mismo/a al reconocer que tus desafíos son válidos y comprensibles.
- Identificación de metas y objetivos realistas para abordar tus dificultades en el hogar, la universidad y otros aspectos de tu vida.
- Implementación de estrategias para mejorar tu productividad académica y establecer un equilibrio entre el estudio y el descanso.
- Mayor sensación de esperanza y optimismo a medida que experimentas pequeños progresos en tus desafíos.

Planificación

Equipo

Valeria	60% Frontend 30% ChatBot 10% Debugging	Valentín	50% Backend 40% Base de Datos 10% Testing	Enzo	60% Frontend 30% Backend 10% Testing	Paolo	50% Backend 40% Base de Datos 10% Testing	Sofía	50% Chatbot 30% Backend 20% Team Lead
CSS, HTML, MySQL, Github, Python		CSS, HTML, Python, MySQL, Github, Markdown		CSS, HTML, Python, MySQL, Github		CSS, HTML, Python, Github		CSS, HTML, Python, MySQL, Github, Markdown	
Cross-Discipline Expertise		Cross-Discipline Expertise		Cross-Discipline Expertise		Cross-Discipline Expertise		Cross-Discipline Expertise	
Datagrip, Monday, Django, Flask Deep Discipline Expertise	AWS, PostgreSQL, Vue, C++	PostgreSQL, Datagrip, Vue Deep Discipline Expertise	AWS, Flask, C++, Algorithms	PostgreSQL, C++, AWS Deep Discipline Expertise	Vue, Flask, node.js, MongoDB	PostgreSQL, AWS Deep Discipline Expertise	Flask, C++	PostgreSQL, C++, Vue, Redis, Jira Deep Discipline Expertise	AWS, Flask, OpenAI, Django

El mejor team! <3

Costos del proyecto

OpenAI: ~\$20/mes

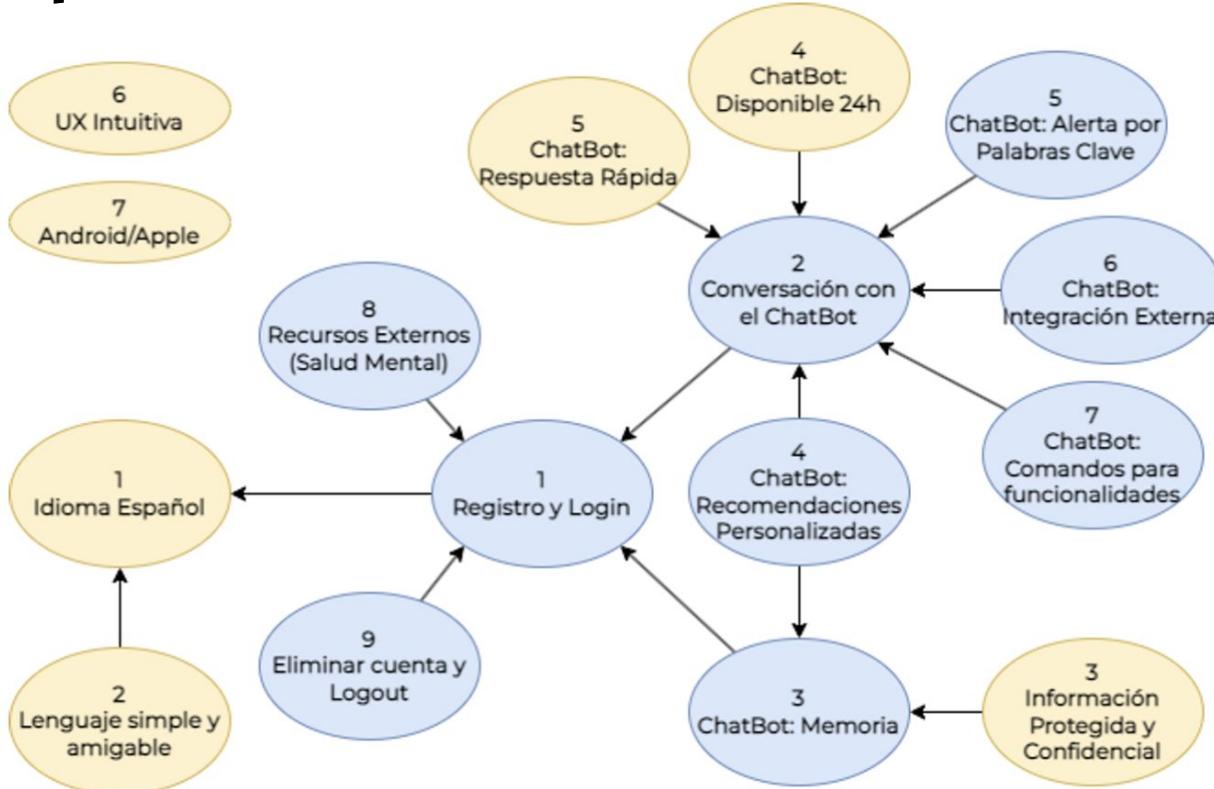
AWS:

- API Gateway: 7 APIs principales x \$3,50 cada millón de llamadas = ~\$24,50/mes
- Amazon S3 Standard: 207 MB x \$0,023 cada GB = ~\$0,004761/mes
- Amazon RDS: ~\$72,80/mes
- Amazon DynamoDB: ~\$38,89/mes
- AWS Lambda: 14 Lambdas principales x ~\$0,20 cada millón de llamadas = ~\$2,80/mes
- Amazon SNS: ~\$0,00 porque solo se usa para conexión entre Lambdas
- Amazon SQS: ~\$0,00 por ser menos de 1 millón de solicitudes por mes
- Amazon Cloudwatch: ~\$2,92/mes

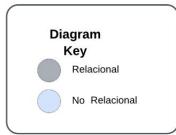
Total: ~\$161,91/mes

¿Cómo funciona Aurora?

Mapa de dependencias



DBM (Updated)



AURORA Diagrama de clases

Grupo 3 | June 23, 2023

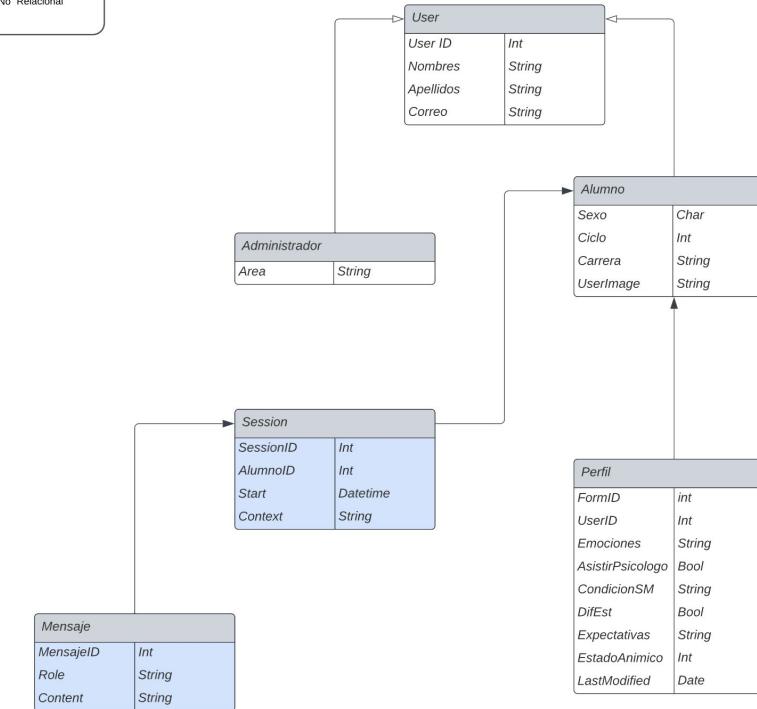


Diagrama de Arquitectura de Soluciones (High Level)

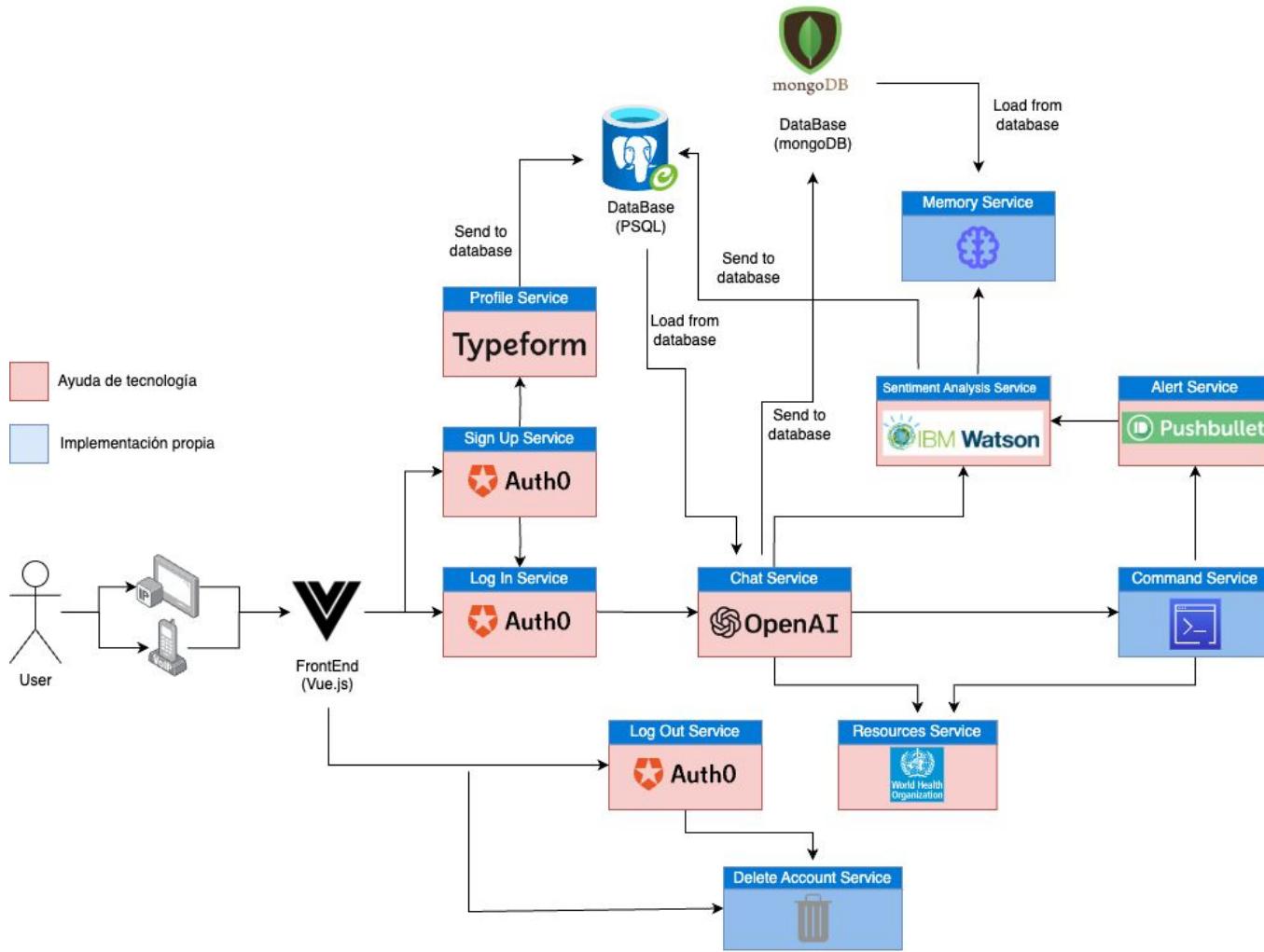


Diagrama de Arquitectura de Soluciones (AWS)

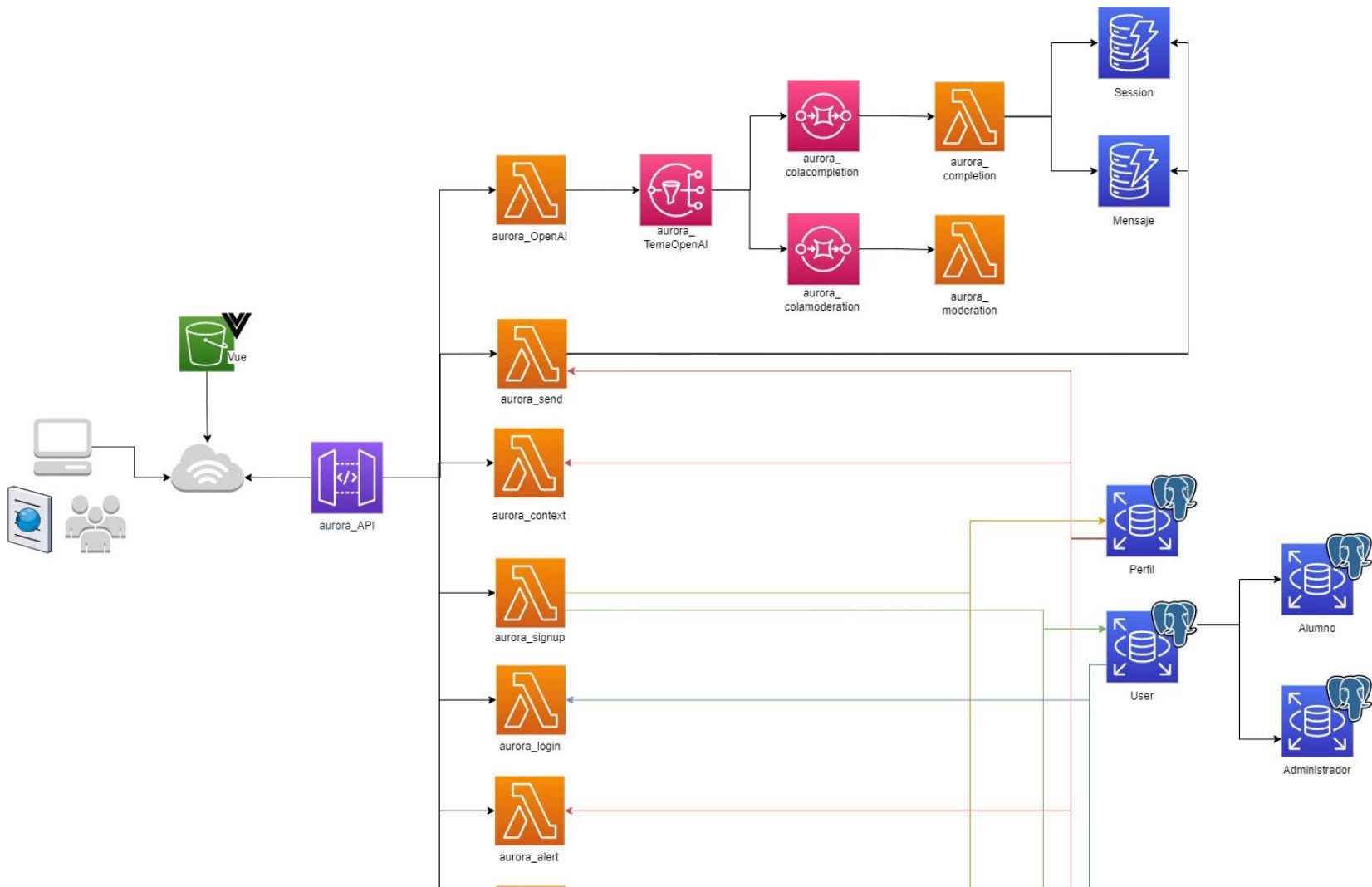
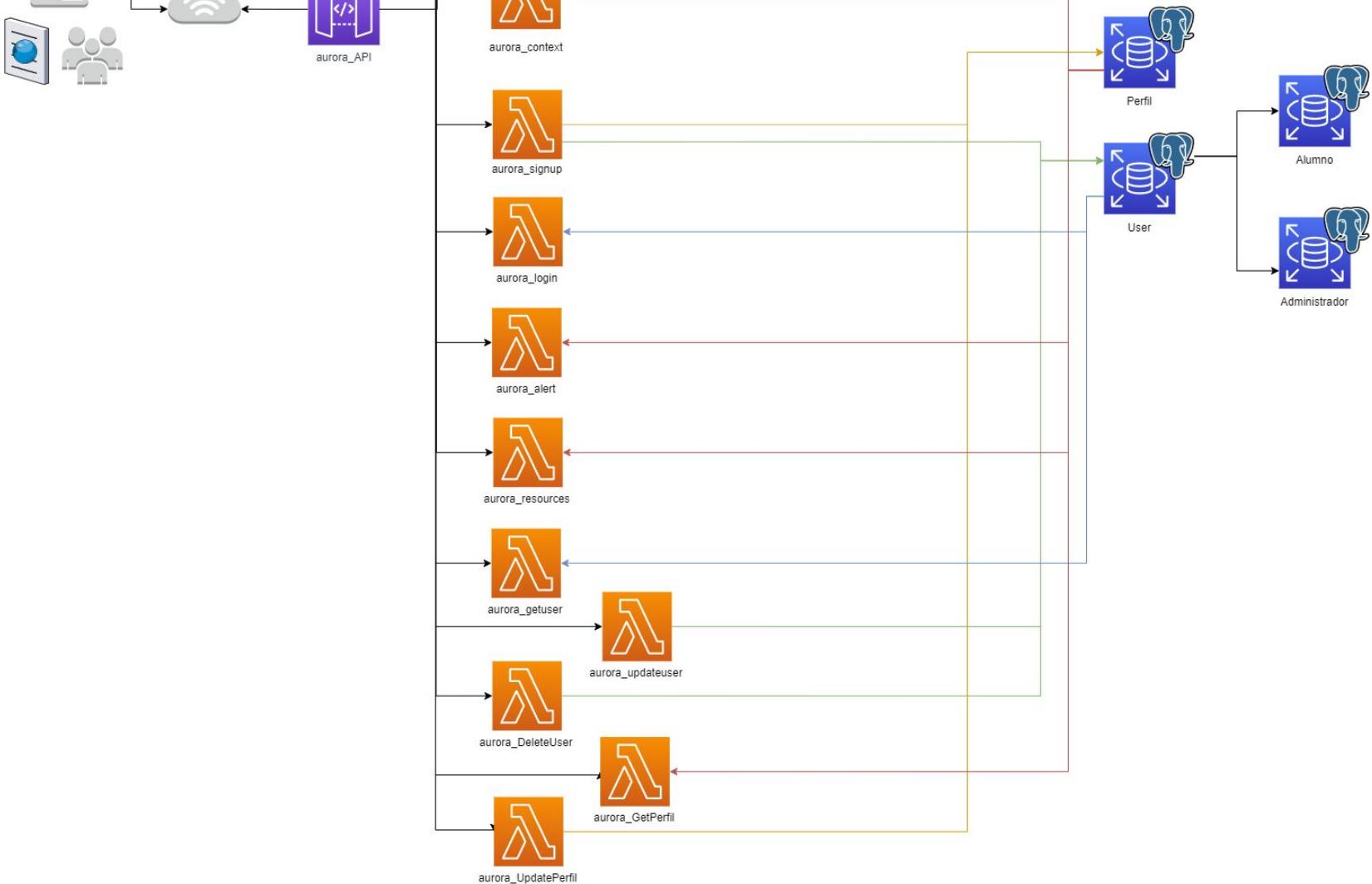


Diagrama de Arquitectura de Soluciones (AWS)



Alcance del producto

Code Coverage

(Comparación)

14/06/2023			30/06/2023			06/07/2023		
Archivo (ejecutable)	Stms	Cover (%)	Archivo (ejecutable)	Stms	Cover (%)	Archivo (ejecutable)	Stms	Cover (%)
registros/__init__.py	217	32%	registros/__init__.py	275	27%	registros/__init__.py	275	27%
mensajeria/__init__.py	98	91%	mensajeria/__init__.py	125	82%	mensajeria/__init__.py	125	82%
alerta/__init__.py	66	91%	alerta/__init__.py	88	78%	alerta/__init__.py	88	78%
-----	-----	-----	recursos/__init__.py	72	76%	recursos/__init__.py	72	76%
test_registros.py	142	100%	test_registros.py	172	100%	test_registros.py	172	100%
test_mensajeria.py	46	100%	test_mensajeria.py	47	100%	test_mensajeria.py	47	100%
test_alerta.py	39	100%	test_alerta.py	40	100%	test_alerta.py	40	100%
-----	-----	-----	test_recursos.py	33	100%	test_recursos.py	33	100%

Code Coverage

(Final)

Archivo (ejecutable)	Stms	Cover (%)
registros/__init__.py	275	83%
mensajeria/__init__.py	125	82%
alerta/__init__.py	88	78%
recursos/__init__.py	72	76%
test_registros.py	172	100%
test_mensajeria.py	47	100%
test_alerta.py	40	100%
test_recursos.py	33	100%

Rendimiento del producto

RENDIMIENTO

(Comparación)

`@app.route("/enviar_recibir/<user_id>", methods = [POST]) {"content": "Hola", "sesionid": 14636 }`

14/06/2023		30/06/2023		06/07/2023	
#Requests	Rendimiento	#Requests	Rendimiento	#Requests	Rendimiento
10	27.2/sec	10	32.2/sec	10	31.5/sec
100	34.3/sec	100	31.5/sec	100	30.3/sec
1000	31.1/sec	1000	28.9/sec	1000	27.6/sec
10000	28.1/sec	10000	27.7/sec	10000	26.7/sec

RENDIMIENTO

(Final)

```
@app.route("/enviar_recibir/<user_id>", methods = [POST]) {"content": "Hola", "sesionid": 14636 }
```

#Requests	Rendimiento
10	30.3/sec
100	32.03/sec
1000	29.2/sec
10000	27.5/sec

Vemos que el producto es estable, ya que mientras más gente lo usaría, responde relativamente a la misma velocidad

Deployment y monitoreo

Deployment (AWS)

Máquina virtual:

EC2

Nombre : a1.metal

vCPU: 16

Memoria: 32 GB

CPU

Picos de demanda por segundo: Utiliza el valor más alto de rendimiento proporcionado en las métricas

Tiempo de procesamiento promedio: Calcula el promedio de los valores de rendimiento proporcionados

Tiempo de ejecución máximo de una función: Este valor depende del límite de tiempo de ejecución establecido para tus funciones en AWS Lambda.

Tamaño de memoria asignada a la función: Este valor se establece al configurar la función en AWS Lambda. Utiliza el tamaño de memoria asignada en gigabytes (GB).

Unidades de capacidad = (Picos de demanda por segundo * Tiempo de procesamiento promedio) / (Tiempo de ejecución máximo de una función)

GB-segundos = Unidades de capacidad * (Tamaño de memoria asignada a la función en GB)

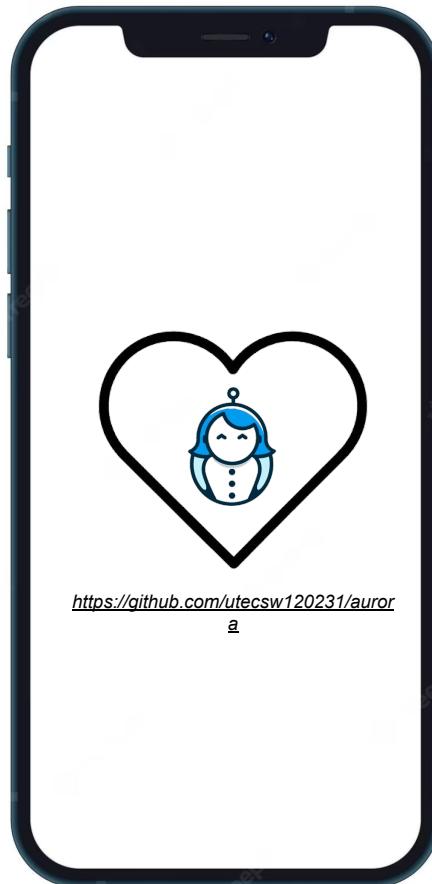
Estas fórmulas se aplican al enfoque serverless de AWS Lambda, donde cada función se ejecuta en una sola CPU virtual y se le asigna una cantidad específica de memoria. AWS Lambda mide el consumo de CPU en unidades de capacidad, que se basan en el tiempo de ejecución y la memoria asignada a la función.

TAMAÑO DE TABLAS EN BYTES

Nombre de tabla	Sistema de Base de Datos	Bytes por atributo	Bytes total
Usuario	PostgreSQL	8+200+200+200+200+200+1	1009
Alumno	PostgreSQL	8+1+8+200	217
Administrador	PostgreSQL	8+200	208
Perfil	PostgreSQL	8+8+200+1+200+1+200+8	626
Sesión	MongoDB	-	617
Messages	MongODB	-	697

DEMO

Código



Cierre



*"Ayuda a las personas a encontrar
una nueva perspectiva en su vida y a
superar los desafíos."*



Enzo Camizan Vidal - enzo.camizan@utec.edu.pe

Paolo Vásquez - paolo.vasquez@utec.edu.pe

Valeria Espinoza - valeria.espinoza@utec.edu.pe

Valentín Quezada - hector.quezada@utec.edu.pe

Sofía García - sofia.garcia@utec.edu.pe