



**Tecnológico
de Monterrey**

Integrantes

Rebeca Monserrat Guevara Lagunas A01651298

Jonatan Hernandez García A01653004

Axel González Carreto A01652775

Marcia Lechuga López A01652732

Desarrollo e implantación de sistemas de software (Gpo 561)

Entregable Ciclo 1

Profesores

Maestra Patricia Escamilla E.

Dra. J. Julieta Noguez Monroy

Maestro Andrés F. Torres Morán

Maestro Jorge Valverde-Rebaza

19 de abril de 2022

Módulo 1. Administración Avanzada de Proyectos	2
Reconsideración de los requerimientos	2
Diseño de la arquitectura de software del proyecto	2
Diseño de componentes del proyecto	3
Casos de Usuario y tareas de ingeniería	4
JIRA	11
Módulo 2. Diseño y Arquitectura	13
Módulo 3. Desarrollo Avanzado de Aplicaciones Web	14
Primera parte	14
Segunda parte	16
Módulo 4. Bases de Datos Avanzadas	18
Staff	18
Recording	29
Instanciar el modelo	20
Usando SDK de AWS DynamoDB	22
Entregables	27
Módulo 5. Calidad del Software	28

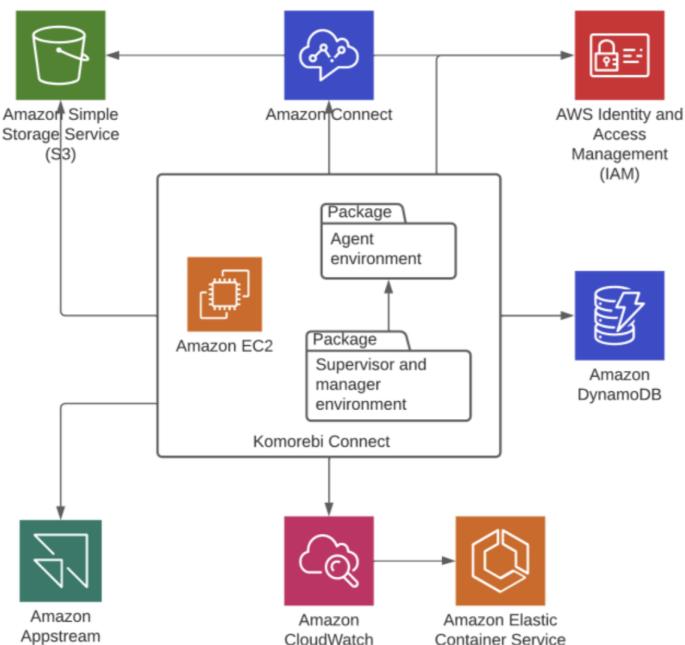
Módulo 1. Administración Avanzada de Proyectos

Reconsideración de los requerimientos

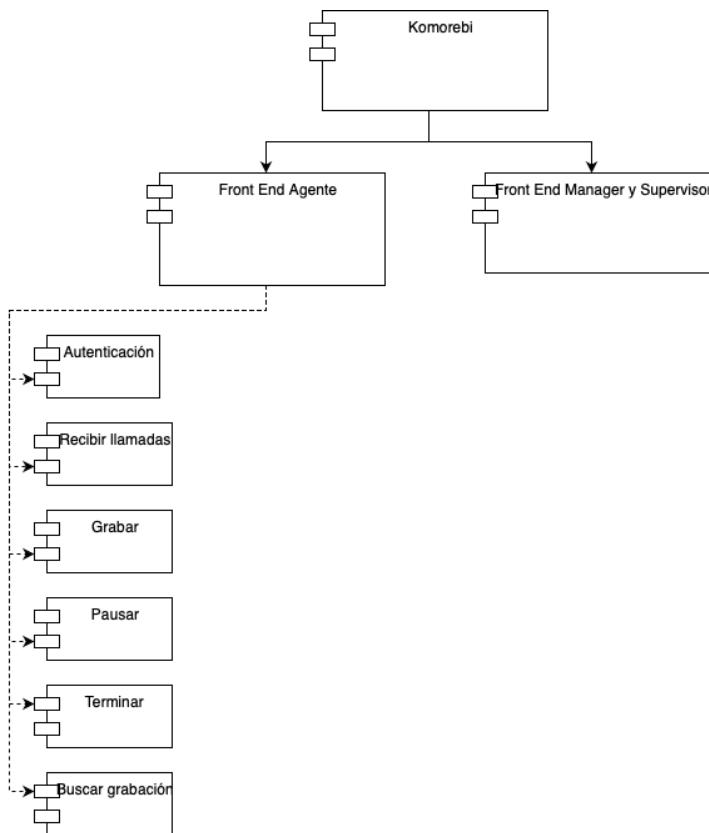
Tomando en cuentas las últimas dos sesiones que se tuvieron con el cliente se modificaron aspectos de dos requerimientos que se tenían. El primero es cuando el agente termina la grabación, específicamente a la encuesta que debe contestar (el requisito con número AG_RF05.1). El cliente pide que se limite las tareas que el agente debe hacer en esta parte por lo que el cuestionario será de pocas preguntas y deberá seleccionar pocos tags.

La otra sugerencia que tuvo el cliente fue el incluir la modalidad de chat en la solución. Para esta sugerencia se tomó en cuenta la grabación de los keystrokes para cumplir con esta solicitud. El transcript de esta grabación será considerado también como el chat.

Diseño de la arquitectura de software del proyecto

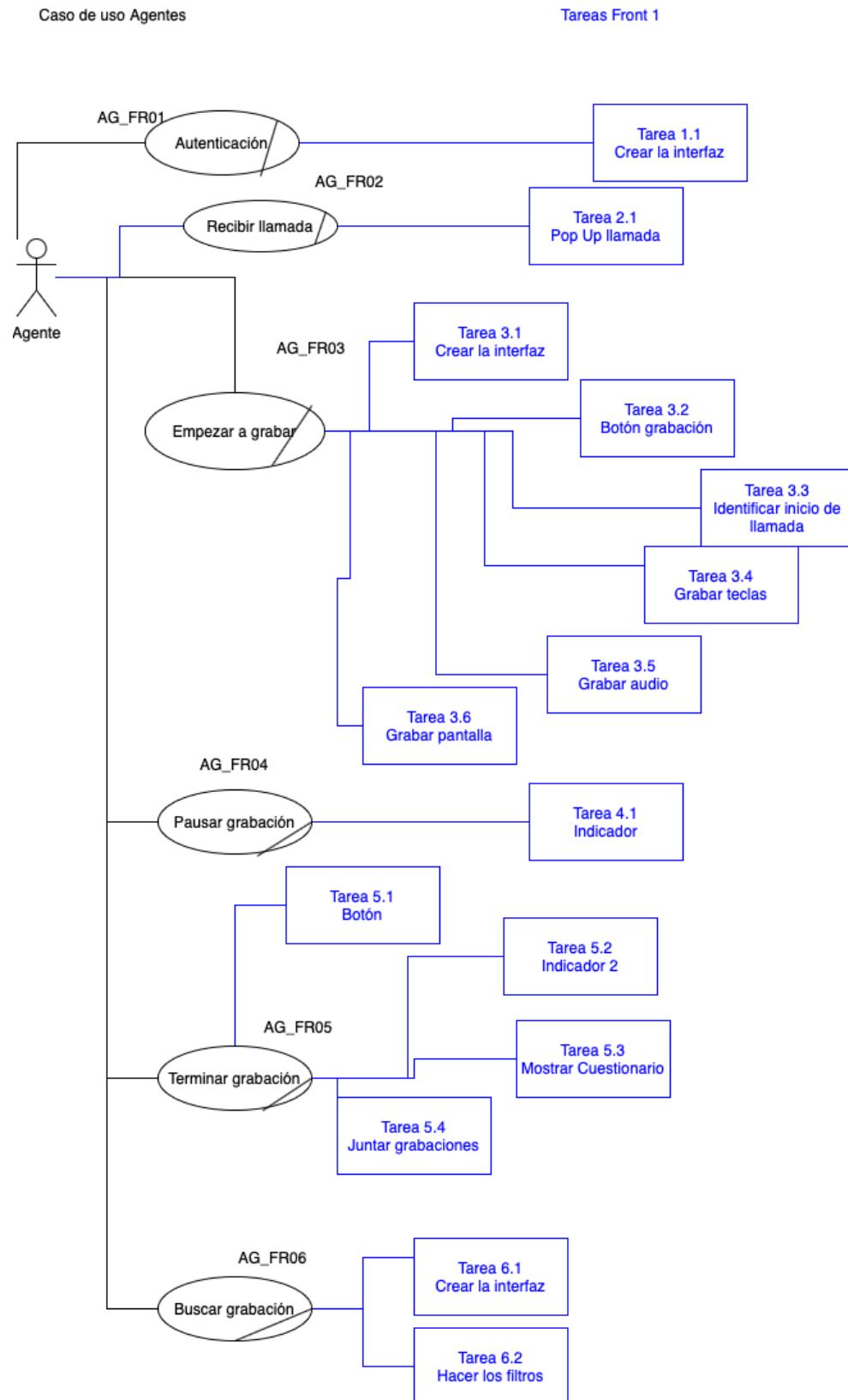


Diseño de componentes del proyecto



Este diseño de componentes está en progreso y solamente se incluyeron los componentes de front end se especificaron los del agente.

Casos de Usuario y tareas de ingeniería



AUTENTICACIÓN

Historia de Usuario	AG_RF01
Número: 1	Nombre: Autenticación
Usuario: Agente	
Modificación de Historia Número:	Iteración asignada:
Prioridad en negocio: Alta	Puntos estimados:
Riesgo en Desarrollo: Bajo	Puntos reales:
Descripción: Crear las interfaces y las conexiones necesarias para que el agente pueda login con sus credenciales autenticadas con la base de datos existente	
Observaciones:	

Tarea	
Número Tarea: 1.1	Número Historia:AG_RF01
Nombre Tarea: Crear la interfaz	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 11 abril 22	Fecha fin: 19 abril 22
Programador responsable: Axel Gonzalez	
Descripción: Con los campos de usuario y contraseña para que el agente pueda entrar a la plataforma necesaria	

RECIBIR LLAMADA

Historia de Usuario	AG_RF02
Número: 2	Nombre: Recibir llamada
Usuario: Agente	
Modificación de Historia Número:	Iteración asignada:
Prioridad en negocio: Alta	Puntos estimados:
Riesgo en Desarrollo: Medio	Puntos reales:

Descripción: Con el API de Amazon Connect se podrán recibir llamadas desde cualquier celular si hay agentes disponibles.
Observaciones:

Tarea	
Número Tarea: 2.1	Número Historia:AG_RF02
Nombre Tarea: Pop Up llamada	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 2/05/22
Programador responsable: Marcia Lechuga	
Descripción: Utilizar el API de Amazon Connect, específicamente la pantalla que despliega la llamada para recibir y terminar llamadas	

EMPEZAR GRABACIÓN

Historia de Usuario	AG_RF03
Número: 3	Nombre: Empezar a grabar
Usuario: Agente	
Modificación de Historia	
Número:	Iteración asignada:
Prioridad en negocio: Alta	Puntos estimados:
Riesgo en Desarrollo: Alto	Puntos reales:
Descripción: El sistema debe comenzar a grabar pantalla, audio y keystrokes cuando el agente acepte la llamada	
Observaciones:	

Tarea	
Número Tarea: 3.1	Número Historia: AG_RF03
Nombre Tarea: Crear la interfaz	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Rebeca Guevara	

Descripción: Crear la interfaz gráfica necesaria.

Tarea	
Número Tarea: 3.2	Número Historia: AG_RF03
Nombre Tarea: Botón grabación	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Jonatan Hernandez	
Descripción: Habilitar un indicador que muestre al agente que la pantalla se ha empezado a grabar	

Tarea	
Número Tarea: 3.3	Número Historia: AG_RF03
Nombre Tarea: Identificar inicio de llamada	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Axel Gonzalez	
Descripción: Reconocer cuando el API de Amazon Connect comience con la llamada	

Tarea	
Número Tarea: 3.4	Número Historia: AG_RF03
Nombre Tarea: Grabar teclas	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable:	
Descripción: Grabar las keystrokes del agente e irlos poniendo en un transcript	

Tarea	
Número Tarea: 3.5	Número Historia: AG_RF03
Nombre Tarea: Grabar audio	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22

Programador responsable: Marcia Lechuga
Descripción: Grabar la conversación del agente con el cliente utilizando el API de Amazon Connect

Tarea	
Número Tarea: 3.6	Número Historia: AG_RF03
Nombre Tarea: Grabar pantalla	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Rebeca Guevara	
Descripción: Grabar toda la pantalla del agente	

PAUSAR GRABACIÓN

Historia de Usuario	AG_RF04
Número: 4	Nombre: Pausar Grabación
Usuario: Agente	
Modificación de Historia	
Número:	Iteración asignada:
Prioridad en negocio:	
Baja	Puntos estimados:
Riesgo en Desarrollo:	
Alto	Puntos reales:
Descripción: Dejar de grabar y hacérselo saber al agente	
Observaciones:	

Tarea	
Número Tarea: 4.1	Número Historia: AG_RF04
Nombre Tarea: Indicador	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Jonatan Hernandez	
Descripción: Que deje de aparecer el indicador de que está grabando	

TERMINAR GRABACIÓN

Historia de Usuario	AG_RF05
Número: 5	Nombre: Terminar grabación
Usuario: Agente	
Modificación de Historia	
Número:	Iteración asignada:
Prioridad en negocio: Alta	Puntos estimados:
Riesgo en Desarrollo: Alto	Puntos reales:
Descripción: Dejar de grabar y hacérselo saber al agente	
Observaciones:	

Tarea	
Número Tarea: 5.1	Número Historia: AG_RF05
Nombre Tarea: Botón	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Axel Gonzalez	
Descripción: Implementar el botón de terminar la llamada en la interfaz gráfica	

Tarea	
Número Tarea: 5.2	Número Historia: AG_RF05
Nombre Tarea: Indicador 2	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Marcia Lechuga	
Descripción:	

Tarea	
Número Tarea: 5.3	Número Historia: AG_RF05
Nombre Tarea: Mostrar Cuestionario	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin:

Programador responsable: Rebeca Guevara
Descripción: Mostrar cuando acabe la grabación una encuesta rápida al agente y se guardará con el botón guardar

Tarea	
Número Tarea: 5.4	Número Historia: AG_RF05
Nombre Tarea: Juntar grabaciones	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 02/05/22
Programador responsable: Jonatan Hernandez	
Descripción: Coordinar las grabaciones de audio con las de video	

BUSCAR GRABACIÓN

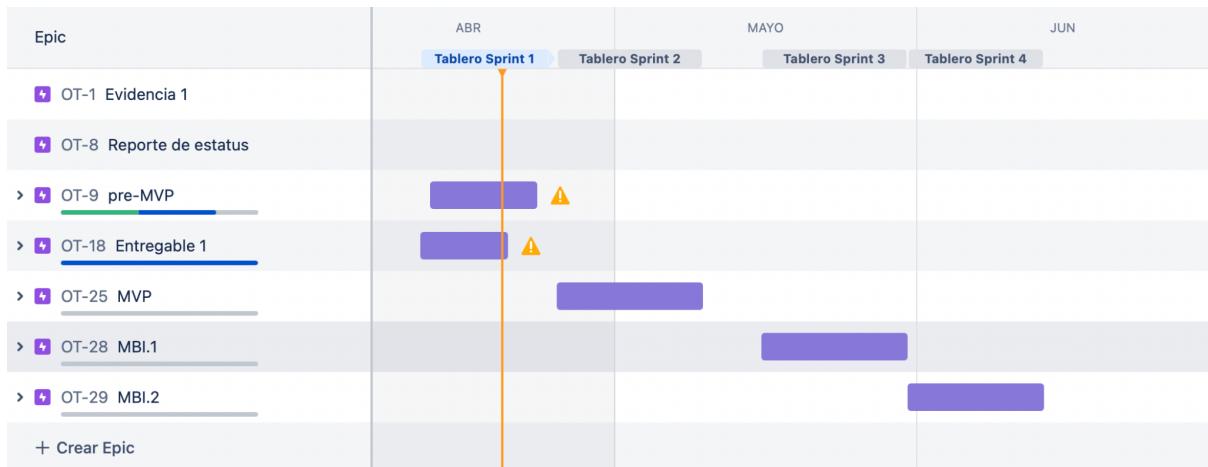
Historia de Usuario	AG_RF06
Número: 6	Nombre: Buscar grabación
Usuario: Agente	
Modificación de Historia	
Número:	Iteración asignada:
Prioridad en negocio: Alta	Puntos estimados:
Riesgo en Desarrollo: Alto	Puntos reales:
Descripción: Crear la interfaces y conexiones necesarias para que le puedan buscar las grabaciones por medio de filtros	
Observaciones:	

Tarea	
Número Tarea: 6.1	Número Historia: AG_RF06
Nombre Tarea: Crear la interfaz	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 11/04/22	Fecha fin: 19/04/22
Programador responsable: Axel Gonzalez	
Descripción: Realizar la interfaz gráfica necesaria	

Tarea	
Número Tarea: 6.2	Número Historia: AG_RF06
Nombre Tarea: Hacer los filtros	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 18/04/22	Fecha fin: 19/04/22
Programador responsable: Axel Gonzalez	
Descripción: Implementar los campos de filtros necesarios	

JIRA

Hoja de ruta



Se organizaron cuatro sprints: El primero acaba esta semana teniendo como EPIC esta entrega, para los siguientes sus EPICs son el MVP, MBI.1 y MBI.2.

Backlog

Tablero Sprint 1 11 abr. – 25 abr. (10 incidencias)		0 0 0	Completar sprint	...
Realizar las tareas por equipo necesarias para la entrega del Ciclo 1				
OT-13	MVP presentación	PRE-MVP	TAREAS POR HACER	...
OT-14	Interfaz (3 pantallas)	PRE-MVP	FINALIZADA	...
OT-15	Creación de base de datos	PRE-MVP	FINALIZADA	...
OT-17	Conexión base y back	PRE-MVP	FINALIZADA	...
OT-19	Tareas de ingeniería	ENTREGABLE 1	FINALIZADA	...
OT-20	Probar AWS llamadas	ENTREGABLE 1	EN CURSO	...
OT-21	Scripts 16	ENTREGABLE 1	FINALIZADA	...
OT-22	Scripts 18	ENTREGABLE 1	FINALIZADA	...
OT-23	Scripts 14	ENTREGABLE 1	FINALIZADA	...

Tareas para el primer sprint, para el segundo sprint donde se presenta el MVP las tareas con parecidas a las tareas de ingeniería de arriba.

Tablero Sprint 2 25 abr. – 9 may. (16 incidencias)		0 0 0	Iniciar sprint	...
Realizar las tareas por equipo necesarias para la entrega del Ciclo 2				
OT-26	MVP	MVP	TAREAS POR HACER	...
OT-31	Crear interfaz login		TAREAS POR HACER	...
OT-32	Pop Up llamada		TAREAS POR HACER	...
OT-33	Crear interfaz grabar		TAREAS POR HACER	...
OT-34	Botón grabacion		TAREAS POR HACER	...
OT-35	Identificar inicio de llamada		TAREAS POR HACER	...
OT-36	Grabar teclas		TAREAS POR HACER	...
OT-37	Grabar audio		TAREAS POR HACER	...
OT-38	Grabar pantalla		TAREAS POR HACER	...

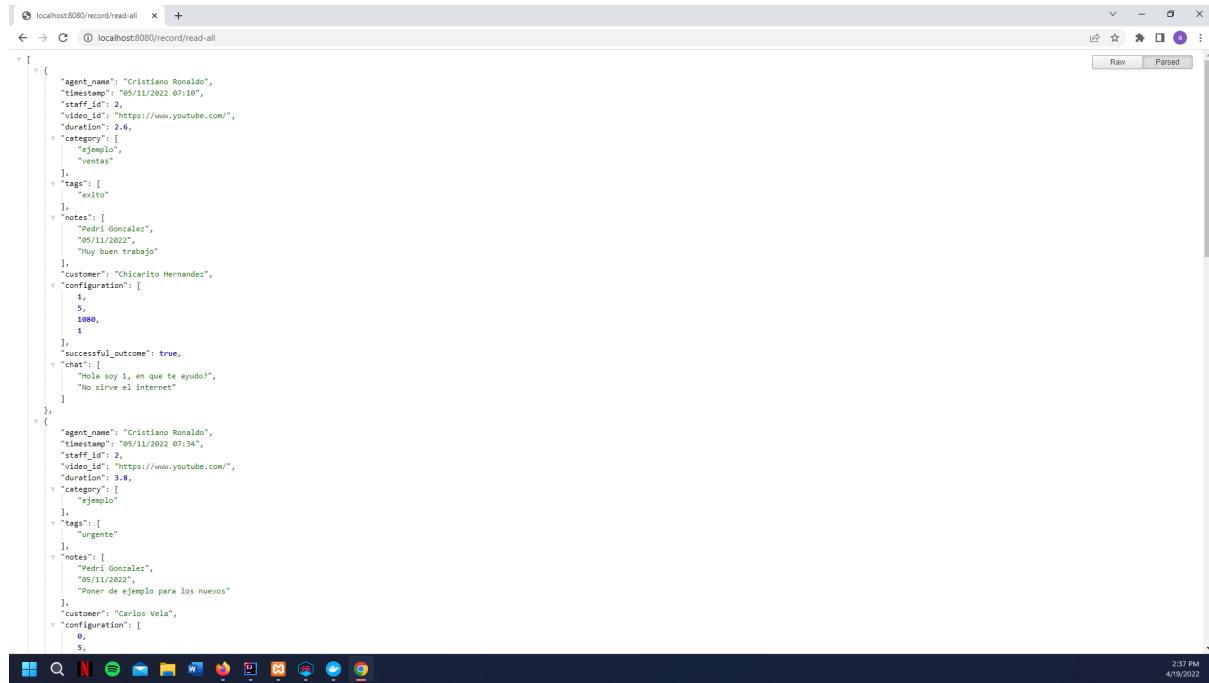
Plan de administración JIRA por equipo para realizar los componentes asignados

<https://oasisoftechnology.atlassian.net/jira/software/projects/OT/boards/1/backlog>

Módulo 2. Diseño y Arquitectura

Transacción Back-end con la Base de datos.

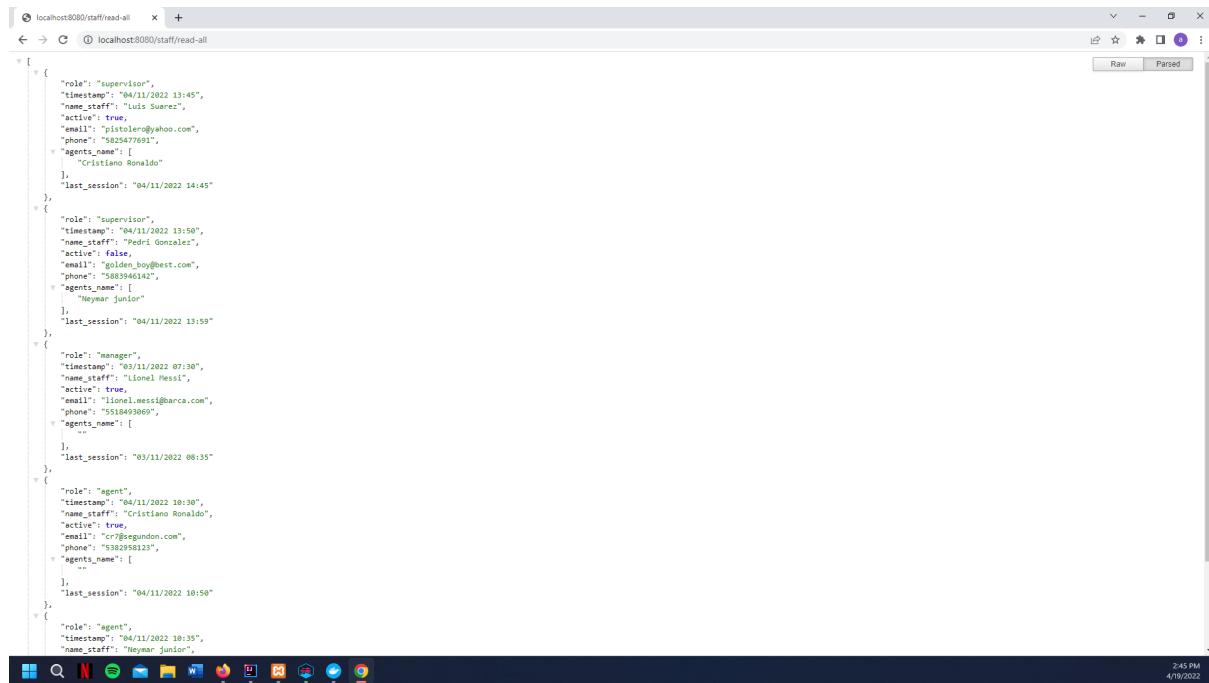
- Scan a tabla RECORD



A screenshot of a browser window titled "localhost:8080/record/read-all". The page displays a JSON array of records. Each record contains fields such as agent_name, timestamp, staff_id, video_id, duration, category, tags, notes, customer, and configuration. The "category" field is consistently set to "ejemplo". The "tags" field includes "exito" or "urgente". The "notes" field often includes "Pedri Gonzalez", "05/11/2022", and "Muy buen trabajo" or "Poner de ejemplo para los nuevos". The "customer" field includes "Chiquito Hernandez" and "Carlos Vela". The "configuration" field includes values like 1, 5, 1000, 1, 0, and 5. The "duration" field varies between 2.6 and 3.8 seconds. The "timestamp" field shows dates from November 5, 2022, at 07:10 to 07:34. The "last_session" field is present in the first record. The browser interface includes tabs for "Raw" and "Parsed", and a status bar indicating 2:37 PM and 4/19/2022.

```
[{"id": 1, "agent_name": "Cristiano Ronaldo", "timestamp": "05/11/2022 07:10", "staff_id": 2, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Chiquito Hernandez", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 1, en que te ayudo?", "time": "05/11/2022 07:10:00"}, {"text": "No sirve el internet", "time": "05/11/2022 07:10:10"}]}, {"id": 2, "agent_name": "Cristiano Ronaldo", "timestamp": "05/11/2022 07:34", "staff_id": 2, "video_id": "https://www.youtube.com/", "duration": 3.8, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "urgente"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Poner de ejemplo para los nuevos"}], "customer": "Carlos Vela", "configuration": [{"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 2, en que te ayudo?", "time": "05/11/2022 07:34:00"}, {"text": "No sirve el internet", "time": "05/11/2022 07:34:10"}]}, {"id": 3, "agent_name": "Luis Suarez", "timestamp": "04/11/2022 13:45", "staff_id": 1, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Luis Suarez", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 3, en que te ayudo?", "time": "04/11/2022 13:45:00"}, {"text": "No sirve el internet", "time": "04/11/2022 13:45:10"}]}, {"id": 4, "agent_name": "Luis Suarez", "timestamp": "04/11/2022 13:50", "staff_id": 1, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Luis Suarez", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 4, en que te ayudo?", "time": "04/11/2022 13:50:00"}, {"text": "No sirve el internet", "time": "04/11/2022 13:50:10"}]}, {"id": 5, "agent_name": "Lionel Messi", "timestamp": "03/11/2022 07:30", "staff_id": 3, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Lionel Messi", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 5, en que te ayudo?", "time": "03/11/2022 07:30:00"}, {"text": "No sirve el internet", "time": "03/11/2022 07:30:10"}]}, {"id": 6, "agent_name": "Lionel Messi", "timestamp": "04/11/2022 10:10", "staff_id": 3, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Lionel Messi", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 6, en que te ayudo?", "time": "04/11/2022 10:10:00"}, {"text": "No sirve el internet", "time": "04/11/2022 10:10:10"}]}, {"id": 7, "agent_name": "Neymar junior", "timestamp": "04/11/2022 10:35", "staff_id": 4, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Neymar junior", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 7, en que te ayudo?", "time": "04/11/2022 10:35:00"}, {"text": "No sirve el internet", "time": "04/11/2022 10:35:10"}]}, {"id": 8, "agent_name": "Neymar junior", "timestamp": "04/11/2022 10:50", "staff_id": 4, "video_id": "https://www.youtube.com/", "duration": 2.6, "category": [{"name": "ejemplo"}, {"name": "veritas"}], "tags": [{"name": "exito"}], "notes": [{"text": "Pedri Gonzalez", "date": "05/11/2022", "content": "Muy buen trabajo"}], "customer": "Neymar junior", "configuration": [{"value": 1}, {"value": 5}, {"value": 1000}, {"value": 1}, {"value": 0}, {"value": 5}], "successful_outcome": true, "chat": [{"text": "Hola soy 8, en que te ayudo?", "time": "04/11/2022 10:50:00"}, {"text": "No sirve el internet", "time": "04/11/2022 10:50:10"}]}]
```

- Scan a tabla STAFF



A screenshot of a browser window titled "localhost:8080/staff/read-all". The page displays a JSON array of staff members. Each staff member is defined by their role, timestamp, name, active status, email, phone number, and a list of agents they manage. The "agents" field lists agents like Cristiano Ronaldo, Luis Suarez, Lionel Messi, and Neymar junior. The "last_session" field indicates the most recent session for each staff member. The browser interface includes tabs for "Raw" and "Parsed", and a status bar indicating 2:45 PM and 4/19/2022.

```
[{"id": 1, "role": "supervisor", "timestamp": "04/11/2022 13:45", "name_staff": "Luis Suarez", "active": true, "email": "luis.suarez@yahoo.com", "phone": "+55543210989", "agents_name": [{"name": "Cristiano Ronaldo"}], "last_session": "04/11/2022 14:45"}, {"id": 2, "role": "supervisor", "timestamp": "04/11/2022 13:50", "name_staff": "Pedri Gonzalez", "active": true, "email": "golden.borg@west.com", "phone": "+58839461342", "agents_name": [{"name": "Neymar junior"}], "last_session": "04/11/2022 13:59"}, {"id": 3, "role": "manager", "timestamp": "03/11/2022 07:30", "name_staff": "Lionel Messi", "active": true, "email": "lionel.messi@barca.com", "phone": "+5518493069", "agents_name": [{"name": "Lionel Messi"}], "last_session": "03/11/2022 08:35"}, {"id": 4, "role": "agent", "timestamp": "04/11/2022 10:10", "name_staff": "Cristiano Ronaldo", "active": true, "email": "cr7@segundon.com", "phone": "+55820958123", "agents_name": [{"name": "Cristiano Ronaldo"}], "last_session": "04/11/2022 10:10"}, {"id": 5, "role": "agent", "timestamp": "04/11/2022 10:35", "name_staff": "Neymar junior", "active": true, "email": "neymar.junior@segundon.com", "phone": "+55820958123", "agents_name": [{"name": "Neymar junior"}], "last_session": "04/11/2022 10:50"}]
```

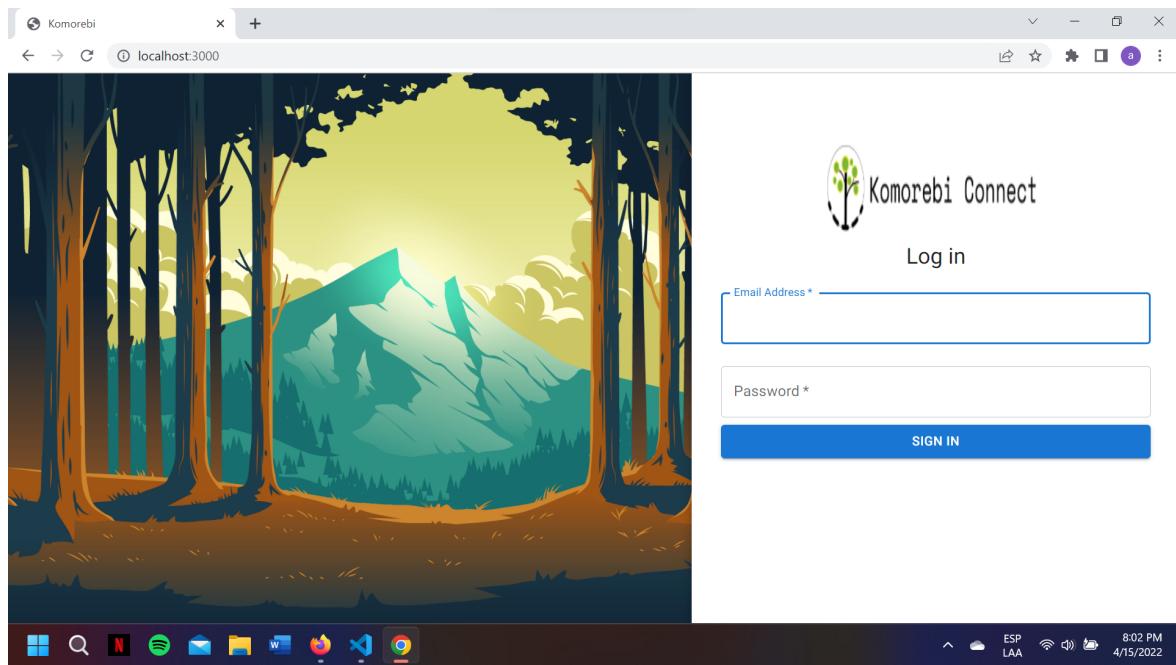
GITHUB: <https://github.com/IIFCAxelentell/komorebi>

Módulo 3. Desarrollo Avanzado de Aplicaciones Web

Primera parte

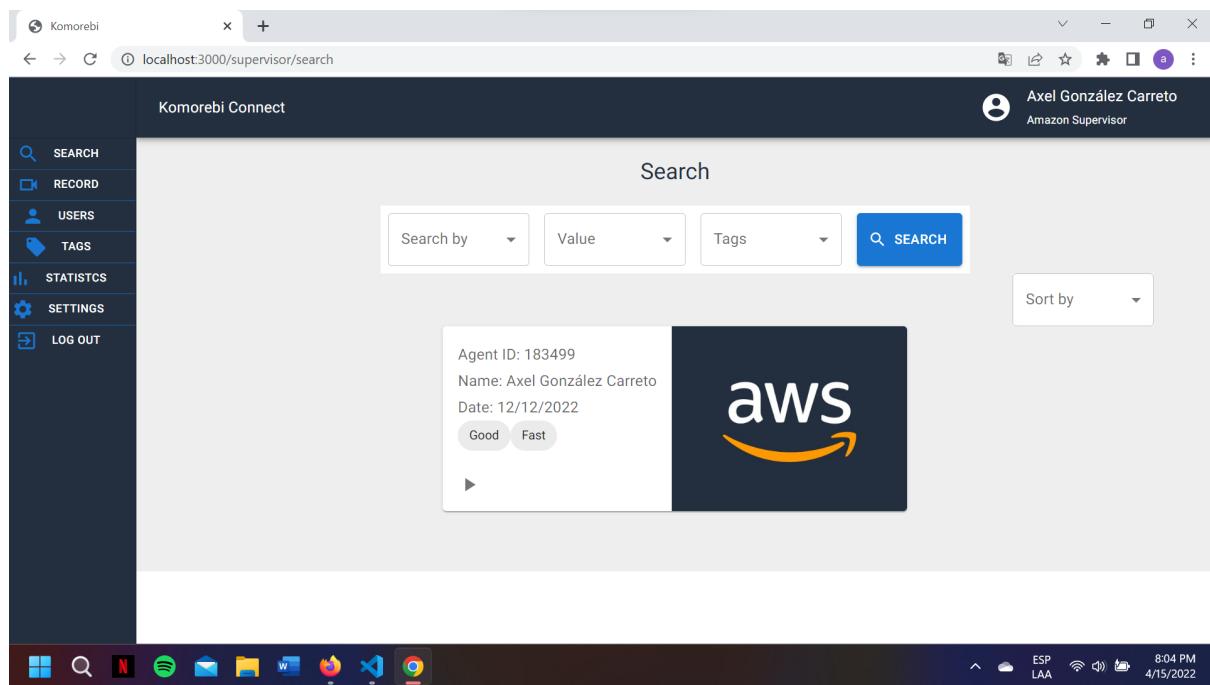
Para esta primera parte, se crearon 3 interfaces completas, las barras de navegación y se hizo que fuera parcialmente navegable. Para su desarrollo se usó Material UI (<https://mui.com/material-ui/getting-started/installation/>).

Se creó la página de inicio (login).

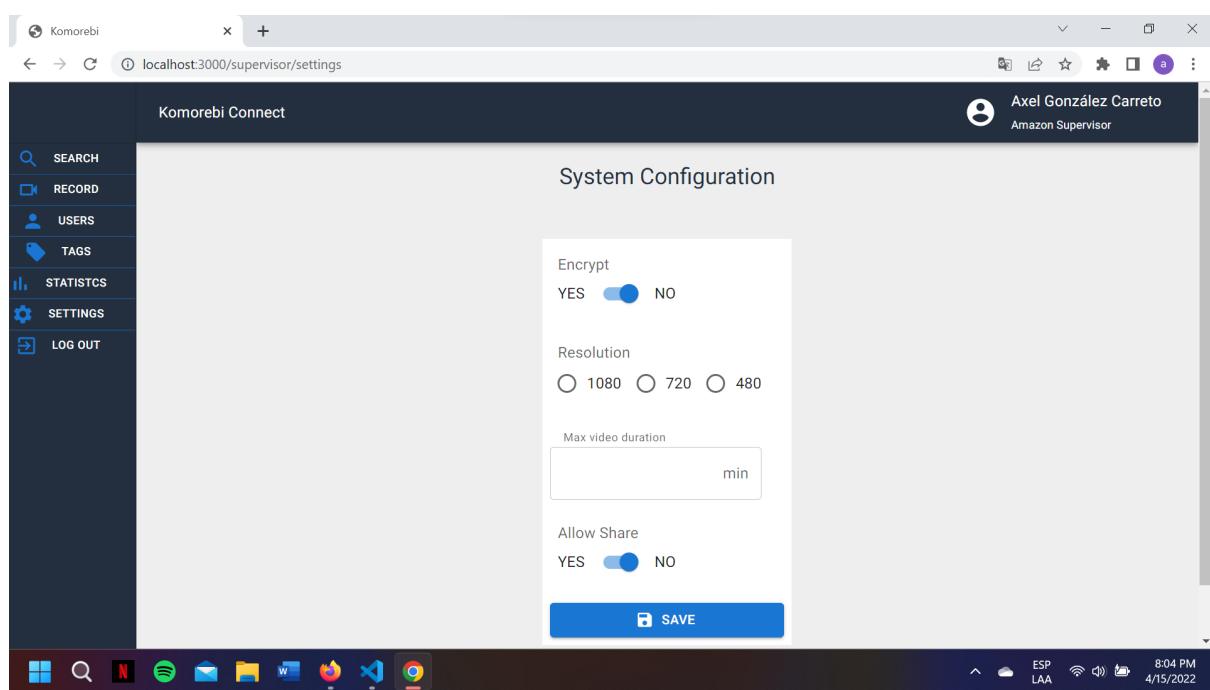


Además, se crearon las interfaces de búsqueda y de configuración del video.

- Search



- Settings



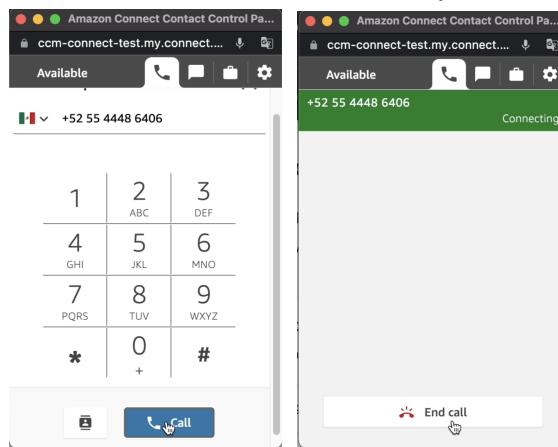
Finalmente, en cuanto a la primera parte, también se hicieron las diferentes sidebar y se hizo que la página fuera navegable.

Se entregará el .zip del proyecto en ReactJS.

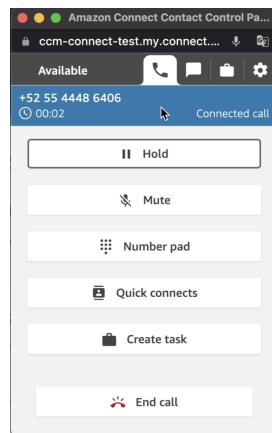
Segunda parte

Utilizando la cuenta de agente realizamos exploramos y nos percatamos de varias cosas, no se tiene acceso para configurar desde esta cuenta, solamente en la de manager. Igualmente solamente el agente puede recibir y realizar llamadas. Tomando en cuenta esto realizamos lo siguiente:

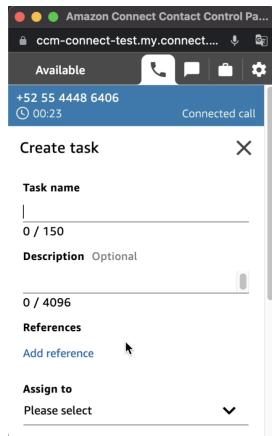
1. Se puede llamar hacia un teléfono desde la plataforma de Amazon Connect



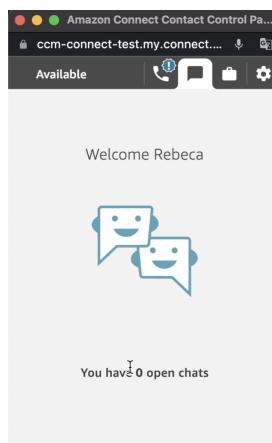
2. Una vez recibida la llamada aparece este menú donde hay varias opciones



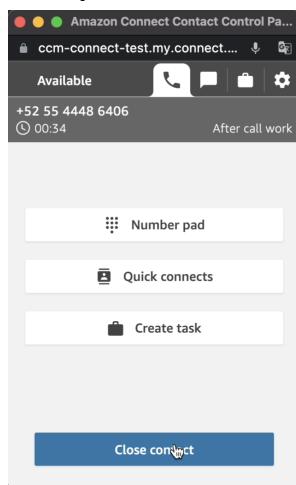
3. Probamos la parte de Create task y te sale lo que parece un cuestionario



4. Igualmente en la parte de arriba aparecen otras opciones como chat, el cual no hemos habilitado todavía



5. Al terminar la llamada se regresará al menú del principio donde permite seleccionar si el agente está disponible o no, al igual esta la opción para llamar, contactos rápidos y crear task.



Link de los videos:

<https://drive.google.com/drive/folders/1q5hcoLdCPYpz3gXaeQR78Ksl0Lb96X31?usp=sharing>

Módulo 4. Bases de Datos Avanzadas

Para esta entrega, se tomaron en cuenta los comentarios realizados en la primera entrega (ciclo 0) y se rediseñaron los modelos, quedando de la siguiente forma.

Staff

Staff			Edit
Primary key attributes ⓘ			
Attribute name	Attribute type	Key type	
role	String	Partition key	
timestamp	String	Sort key	
Other attributes ⓘ			
Attribute name	Attribute type		
name_staff	String		
active	Boolean		
email	String		
phone	String		
agents_name	String Set		
last_session	String		

Primary key		Attributes					
Partition key: role	Sort key: timestamp	name_staff	active	email	phone	agents_name	last_session
manager	03/11/2022 07:30	Lionel Messi	true	lionel.messi@barca.com	5518493069	["]	03/11/2022 08:35
		Cristiano Ronaldo	true	cr7@segundon.com	5382958123	["]	04/11/2022 10:50
agent	04/11/2022 10:30	Neymar junior	false	sombra@gmail.com	5938608759	["]	04/11/2022 10:38
		Luis Suarez	true	pistolero@yahoo.com	5825477691	["Cristiano Ronaldo"]	04/11/2022 14:45
supervisor	04/11/2022 13:45	Pedri Gonzalez	false	golden_boy@best.com	5883946142	["Neymar junior"]	04/11/2022 13:59

Ahora la clave de partición es el rol, la clave de ordenamiento es la fecha de creación (timestamp); además, se cuenta con el nombre del empleado (string), si está activo o no (bool), su correo electrónico (string), telefono (string), un set de nombres de agentes asignados (si no es un supervisor, estará vacío; de lo contrario, contará con el nombre de los agentes a su cargo) (string set) y su última fecha de inicio de sesión (string).

Recording

Recording			
			Edit
Primary key attributes ⓘ			
Attribute name		Attribute type	Key type
agent_name		String	Partition key
timestamp		String	Sort key
Other attributes ⓘ			
Attribute name		Attribute type	
staff_id		Number	
video_id		String	
duration		Number	
category		String Set	
tags		String Set	
notes		List	
customer		String	
configuration		List	
successful_outcome		Boolean	
chat		List	

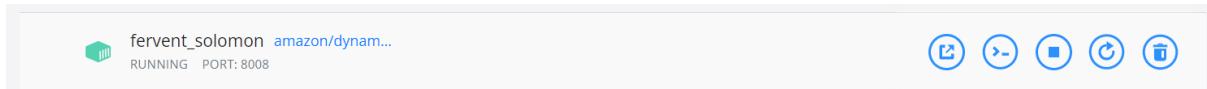
Primary key		Attributes									
Partition key: agent_name	Sort key: timestamp										
Cristiano Ronaldo	05/11/2022 07:10	2	https://www.youtube.com/	2.6	["ventas", "ejemplo"]	["exito"]	[{"S": "Pedri Gonzalez"}, {"S": "05/11/2022"}, {"S": "Muy bien trabajo"}]	Chicarito Hernandez	[{"N": "1"}, {"N": "5"}, {"N": "1080"}, {"N": "1"}]	true	[{"S": "Hola soy 1, en que te ayudo?"}, {"S": "No sirve el internet"]]
	05/11/2022 07:34	2	video_id	duration	category	tags	notes	customer	configuration	successful_outcome	chat
			https://www.youtube.com/	3.8	["ejemplo"]	["urgente"]	[{"S": "Pedri Gonzalez"}, {"S": "05/11/2022"}, {"S": "Poner de ejemplo para los nuevos"]]	Carlos Vela	[{"N": "0"}, {"N": "5"}, {"N": "1080"}, {"N": "0"}]	true	[{"S": "Hola soy 2, en que te ayudo?"}, {"S": "No puedo ver la tele"]]
	05/11/2022 10:02	2	staff_id	video_id	duration	category	tags	notes	customer	configuration	successful_outcome
			https://www.youtube.com/	5	["tipo 2", "tipo 3"]	["prueba"]	[{"S": "Luis Suarez"}, {"S": "05/11/2022"}, {"S": "Este video es de prueba"}]	Hirvin Lozano	[{"N": "0"}, {"N": "5"}, {"N": "120"}, {"N": "1"}]	false	[{"S": "Hola soy 3, en que te ayudo?"}, {"S": "Quiero ponerte a prueba"}]
	05/11/2022 07:00	3	staff_id	video_id	duration	category	tags	notes	customer	configuration	successful_outcome
Neymar junior	05/11/2022 08:46	3	https://www.youtube.com/	2.2	["tipo 2", "tipo 3", "tipo 4"]	["completo", "exito"]	[{"S": "Luis Suarez"}, {"S": "05/11/2022"}, {"S": "Una actuación buena"}]	Tecatito Corona	[{"N": "0"}, {"N": "5"}, {"N": "1080"}, {"N": "1"}]	false	[{"S": "Hola soy 4, en que te ayudo?"}, {"S": "Quiero comprar un perfume"}]
			staff_id	video_id	duration	category	tags	notes	customer	configuration	successful_outcome
			https://www.youtube.com/	3.7	["tipo 1"]	["urgente", "f alla"]	[{"S": "Luis Suarez"}, {"S": "05/11/2022"}, {"S": "Messi es mejor que Péñalo"}]	Andres Guardado	[{"N": "1"}, {"N": "5"}, {"N": "1080"}, {"N": "1"}]	true	[{"S": "Hola soy 5, en que te ayudo?"}, {"S": "No sirve el ref"}]

Ahora la clave de partición es el nombre del agente, la clave de ordenamiento es la fecha de creación del video (timestamp); además se tiene el id del empleado (number), la url del video (string), un set de categorías (string set), otro de etiquetas (string set), se tiene una lista de notas con el formato nombre (string), fecha (string) y texto (string), se cuenta con el nombre del cliente (string), con una lista de configuración del video con el formato encriptar video (0 no, 1 sí), duración máxima (number), resolución del video (number) y compartir video (0 no, 1 sí), también se tiene si la salida fue exitosa (bool) y finalmente una última lista del chat.

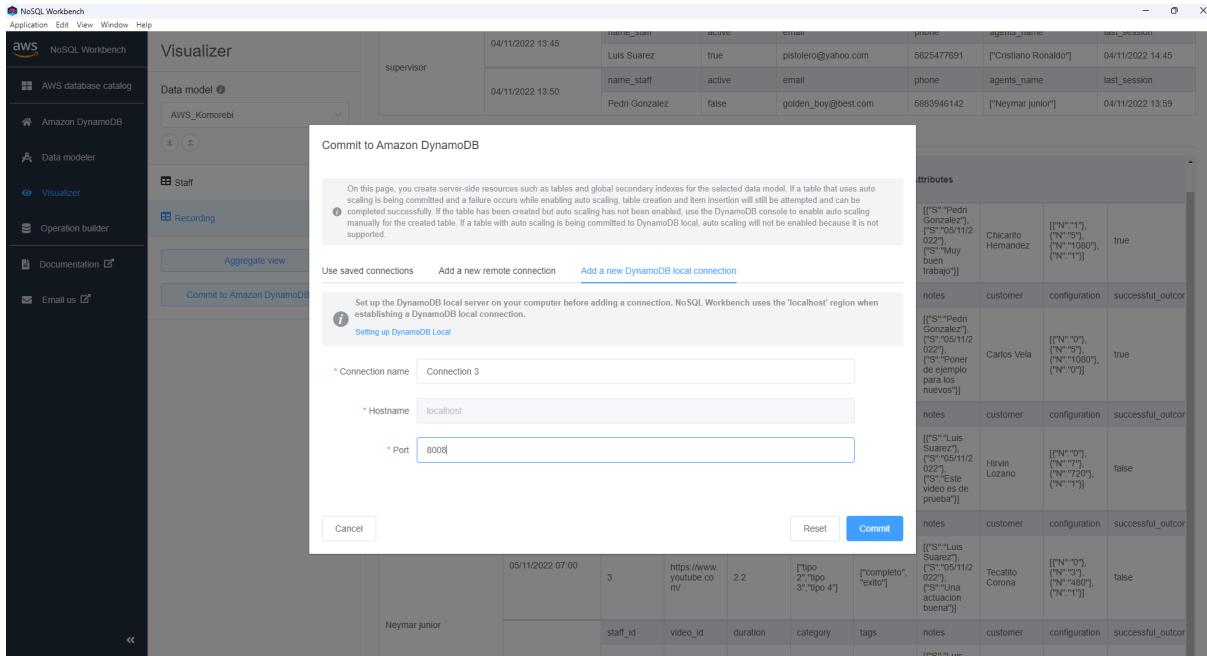
Instanciar el modelo

Una vez modificadas las tablas y con algunos datos para prueba, podemos proceder a instalar nuestro modelo.

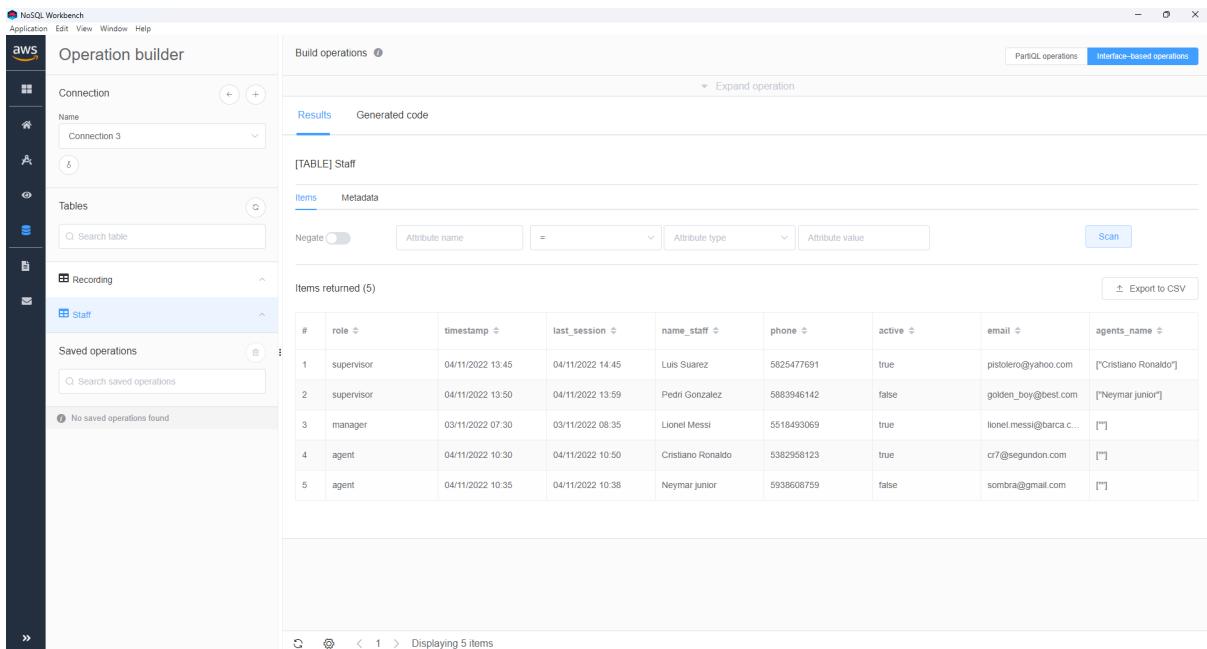
1. Activar en Docker la imagen/contenedor de amazon/dynamodb local.



2. En NoSQL WorkBench ir al apartado Visualizer, Commit to Amazon DynamoDB a la sección Add a new DynamoDB local connection. Especificar puerto (debe coincidir con el Docker) y darle commit.



3. Esto nos creará una instancia, así mismo, nos genera nuestras tablas con nuestros datos previamente ingresados.



Usando SDK de AWS DynamoDB

Usaremos el SDK para crear un programa en Java que sea capaz de enlistar las tuplas almacenadas en cada tabla y para ejecutar un query para obtener sólo algunas tuplas con base en un predicado. Se creara el código en NoSQL Workbench para poder hacer el programa en IntelliJ, se debe especificar el puerto, access key id y secret access key en el código generado por NoSQL Workbench.

- Scan en tabla Staff

En NoSQL Workbench:

The screenshot shows the NoSQL Workbench application window. On the left, there's a sidebar with 'Operation builder' and sections for 'Connection' (set to 'Connection 3'), 'Tables' (listing 'Recording' and 'Staff'), and 'Saved operations'. The main area is titled 'Build operations' and shows a 'Scan' operation on the 'Staff' table. Below this, under 'Results', a table displays 5 items returned from the scan. The table has columns: #, role, last_session, name_staff, phone, active, email, timestamp, and agents_name. The data is as follows:

#	role	last_session	name_staff	phone	active	email	timestamp	agents_name
1	supervisor	04/11/2022 14:45	Luis Suarez	5825477691	true	pistolero@yahoo.com	04/11/2022 13:45	["Cristiano Ronaldo"]
2	supervisor	04/11/2022 13:59	Pedri Gonzalez	5883946142	false	golden_boy@best.com	04/11/2022 13:50	["Neymar junior"]
3	manager	03/11/2022 08:35	Lionel Messi	5516493069	true	lionel.messi@barca.c...	03/11/2022 07:30	[]
4	agent	04/11/2022 10:50	Cristiano Ronaldo	5382958123	true	cr7@segundon.com	04/11/2022 10:30	[]
5	agent	04/11/2022 10:38	Neymar junior	5938608759	false	sombra@gmail.com	04/11/2022 10:35	[]

En Java (IntelliJ):

```

22:18:58.085 [main] DEBUG com.amazonaws.requestId - x-amzn-RequestID: 1e71b266-e1a-40ec-8e40-c49fe4853cb8
role S=[supervisor]
last_session S=04/11/2022 14:45
name_staff S=[Luis Suarez]
phone S=[+5625477691]
active
email S=[pistole@ yahoo.com]
timestamp S=04/11/2022 13:45
agents_name SS=[{Cristiano Ronaldo}]
role S=[supervisor]
last_session S=04/11/2022 13:59
name_staff S=[Pedro Gonzalez]
phone S=[+5683946142]
active
email S=[golden_boy@ best.com]
timestamp S=04/11/2022 13:50
agents_name SS=[{Neymar junior}]
role S=[manager]
last_session S=05/11/2022 08:35
name_staff S=[Lionel Messi]
phone S=[+591849569]
active
email S=[lionel.messi@ barca.com]
timestamp S=05/11/2022 07:30
agents_name SS=[{}]
role S=[agent]
last_session S=04/11/2022 10:50
name_staff S=[Cristiano Ronaldo]
phone S=[+5582958123]
active
email S=[cr7@ segundon.com]
timestamp S=04/11/2022 10:30
agents_name SS=[{}]
role S=[agent]
last_session S=04/11/2022 10:38
name_staff S=[Neymar junior]
phone S=[+5938608759]
active
email S=[sombra@gmail.com]
timestamp S=04/11/2022 10:35
agents_name SS=[{}]
  
```

- Query y scan en tabla Staff con condiciones

1. Query role = agent

En NoSQL Workbench:

The screenshot shows the AWS NoSQL Workbench interface. On the left, there's a sidebar with 'aws' selected, showing 'Operation builder', 'Connection' (Connection 3), 'Tables' (Recording, Staff), and 'Saved operations'. The main area has 'Build operations' tabs for 'Partial operations' (selected) and 'Interface-based operations'. Under 'Data plane operations', it's set to 'Query' and 'Table' is 'Staff'. The 'Partition key' is set to 'agent'. In the 'Results' tab, it says 'Items returned (2)'. The table shows:

#	role	last_session	name_staff	phone	active	email	timestamp	agents_name
1	agent	04/11/2022 10:50	Cristiano Ronaldo	+5582958123	true	cr7@segundon.com	04/11/2022 10:30	[]
2	agent	04/11/2022 10:38	Neymar junior	+5938608759	false	sombra@gmail.com	04/11/2022 10:35	[]

En Java (IntelliJ):

```

22:41:12.790 [main] DEBUG org.apache.http.wire - http-outgoing-0 <> "x-amzn-RequestId: 21d02a09-4321-4756-bb8c-b22f54d22b25[\r][\n]"
22:41:12.790 [main] DEBUG org.apache.http.wire - http-outgoing-0 <> "Content-Length: 531[\r][\n]"
22:41:12.790 [main] DEBUG org.apache.http.wire - http-outgoing-0 <> "[\r][\n]"
22:41:12.790 [main] DEBUG org.apache.http.wire - http-outgoing-0 <> "Server: Jetty(9.4.18.v20190429)[\r][\n]"
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> "HTTP/1.1 200 OK"
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> Content-Type: application/x-amz-json-1.0
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> x-amz-crc32: 603922965
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> x-amzn-RequestId: 21d02a09-4321-4756-bb8c-b22f54d22b25
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> Content-Length: 531
22:41:12.793 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> Server: Jetty(9.4.18.v20190429)
22:41:12.800 [main] DEBUG com.amazonaws.retry.ClockSkewAdjuster - Reported server date (from 'Date' header): Tue, 12 Apr 2022 05:41:12 GMT
22:41:12.829 [main] DEBUG org.apache.http.impl.cookie.PoolingHttpClientConnectionManager - Connection [id: 0][route: {}->http://localhost:8080] can be kept alive for 60.0 seconds
22:41:12.829 [main] DEBUG org.apache.http.impl.DefaultManagedHttpClientConnection - http-outgoing-0: set socket timeout to 0
22:41:12.829 [main] DEBUG org.apache.http.impl.cookie.PoolingHttpClientConnectionManager - Connection released: [id: 0][route: {}->http://localhost:8080][total available: 1; route allocated: 1 of 50; total allocated: 1 of 50]
22:41:12.830 [main] DEBUG com.amazonaws.request - Received successful response: 200, AWS Request ID: 21d02a09-4321-4756-bb8c-b22f54d22b25
22:41:12.830 [main] DEBUG com.amazonaws.request - Received successful response: 200, AWS Request ID: 21d02a09-4321-4756-bb8c-b22f54d22b25
Process finished with exit code 0

```

Build completed successfully in 2 sec, 957 ms (moments ago)

556 CRLF UTF-8 4 spaces

2. Scan active = true

En NoSQL Workbench:

Operation builder

Connection: Connection 3

Tables: Staff

Filter expression: active = true

Results (3)

#	role	last_session	name_staff	phone	active	email	timestamp	agents_name
1	supervisor	04/11/2022 14:45	Luis Suarez	5625477691	true	pistolero@yahoo.com	04/11/2022 13:45	["Cristiano Ronaldo"]
2	manager	03/11/2022 08:35	Lionel Messi	5518493069	true	lionel.messi@barca.c...	03/11/2022 07:30	[""]
3	agent	04/11/2022 10:50	Cristiano Ronaldo	5382958123	true	cr7@segundon.com	04/11/2022 10:30	[""]

En Java (IntelliJ):

The screenshot shows an IDE interface with multiple tabs open, all related to AmazonDynamoDBScanGeneratedCodeStaffActiveTrue.java. The code in the editor is a Java application that performs a scan operation on an Amazon DynamoDB table. The log output shows the execution of the application, including connection details, retries, and the successful execution of the scan command.

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ConexionDynamo0561 - AmazonDynamoDBScanGeneratedCodeStaffActiveTrue.java
```

```
ConexionDynamo0561 src main java mx_gonzalez_ConexionDynamo0561 AmazonDynamoDBScanGeneratedCodeStaffActiveTrue printenv
```

```
AmazonDynamoDBScanGeneratedCodeRecording.java AmazonDynamoDBQueryGeneratedCodeStaffRoleAgent.java AmazonDynamoDBScanGeneratedCodeStaffActiveTrue.java AmazonDynamoDBScanGeneratedCodeStaffActiveTrue.java
```

```
Project Run AmazonDynamoDBScanGeneratedCodeStaffActiveTrue
```

```
22:51:00.778 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> x-amz-crc32: 346454654
22:51:00.778 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> x-amz-RequestID: d09f1376-cdca-43cc-8ce3-48f87cb3d756
22:51:00.778 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> Content-Length: 384
22:51:00.778 [main] DEBUG org.apache.http.headers - http-outgoing-0 <> Server: Jetty(9.4.18.v20190429)
22:51:00.786 [main] DEBUG org.apache.http.impl.cookieookieookieookieookeMainClientExec - Connection can be kept alive for 00000 MILLISECONDS
22:51:00.786 [main] DEBUG org.apache.http.impl.cookieookieookeClockSkewAdjuster - Reported server date (from 'Date' header): Tue, 12 Apr 2022 03:50:59 GMT
22:51:00.815 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager - Connection [id: 0][route: {}->http://localhost:8008] can be kept alive for 60.0 seconds
22:51:00.816 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager - Connection released: [id: 0][route: {}->http://localhost:8008][total available: 1; route allocated: 1 of 50; total allocated: 1 of 50]
22:51:00.817 [main] DEBUG com.amazonaws.request - Received successful response: 200 AWS Request ID: d09f1376-cdca-43cc-8ce3-48f87cb3d756
22:51:00.817 [main] DEBUG com.amazonaws.requestId - x-amz-RequestId: d09f1376-cdca-43cc-8ce3-48f87cb3d756
```

```
role S-[supervisor]
last_session S-[04/11/2022 14:45]
name_staff S-[Luis Suarez]
phone S-[582547691]
active 800L[true]
email S-[pistolero@yahoo.com]
timestamp S-[04/11/2022 13:45]
agents_name S-[[Cristiano Ronaldo]]
role S-[manager]
last_session S-[03/11/2022 08:55]
name_staff S-[Lionel Messi]
phone S-[5518493009]
active 800L[true]
email S-[Lionel_messi@barca.com]
timestamp S-[03/11/2022 07:50]
agents_name S-[[ ]]
role S-[agent]
last_session S-[04/11/2022 10:50]
name_staff S-[Cristiano Ronaldo]
phone S-[5382958123]
active 800L[true]
email S-[cr7@segundodon.com]
timestamp S-[04/11/2022 10:30]
agents_name S-[[ ]]
Scan successful.
```

```
Process finished with exit code 0
```

- Scan en tabla Recording

En NoSQL Workbench:

En Java (IntelliJ):

```

22:33:32.975 [main] DEBUG org.apache.http.impl.conn.PoolingHttpClientConnectionManager - Connection released: [id: 0][route: {}->http://localhost:8000][total available: 1; route allocated: 1 of 50; total allocated: 1 of 50]
22:33:32.978 [main] DEBUG com.amazonaws.request - Received response: 200, AWS Request ID: 7c1c9e6a-5d70-4bff-a0df-a2aa22e5c013
duration N<[2.0]
notes L<[[{"S": "Pedro Gonzalez,"}, {"S": "05/11/2022,"}, {"S": "Muy buen trabajo,"}]] 
agent_name S<[Cristiano Ronaldo]
configuration L<[[{"N: 1,"}, {"N: 5,"}, {"N: 1080,"}, {"N: 1,}]] 
chat L<[{"S": "Hola soy 1, en que te ayudo?,"}, {"S": "No sirve el internet,"}]] 
staff_id N<[2]
category SS<[{"S": "empleado, ventas"}]
timestamp S<[05/11/2022 07:10]
video_id S<[https://www.youtube.com/]
tags S<[{"S": "[exito]"}]
customer S<[chicarito Hernandez]
successful_outcome
duration N<[3.8]
notes L<[[{"S": "Pedro Gonzalez,"}, {"S": "05/11/2022,"}, {"S": "Poner de ejemplo para los nuevos,"}]] 
agent_name S<[Cristiano Ronaldo]
configuration L<[[{"N: 0,"}, {"N: 5,"}, {"N: 1080,"}, {"N: 0,}]] 
chat L<[{"S": "Hola soy 2, en que te ayudo?,"}, {"S": "No puedo ver la tele,"}]] 
staff_id N<[2]
category SS<[{"S": "empleado"}]
timestamp S<[05/11/2022 07:34]
video_id S<[https://www.youtube.com/]
tags S<[{"S": "[urgente]"}]
customer S<[carlos Vela]
successful_outcome
duration N<[3.8]
notes L<[[{"S": "Luis Suarez,"}, {"S": "05/11/2022,"}, {"S": "Este video es de prueba,"}]] 
agent_name S<[Cristiano Ronaldo]
configuration L<[[{"N: 0,"}, {"N: 7,"}, {"N: 720,"}, {"N: 1,}]] 
chat L<[{"S": "Hola soy 3, en que te ayudo?,"}, {"S": "Quiero ponerle a prueba,"}]] 
staff_id N<[2]
category SS<[{"S": "tipo 2, tipo 3"}]
timestamp S<[05/11/2022 10:02]
video_id S<[https://www.youtube.com/]
tags S<[{"S": "[prueba]"}]
customer S<[Kirkvain Lozano]
successful_outcome
duration N<[2.2]

```

Build completed successfully in 3 sec, 47 ms (moments ago)

- Scan en tabla Recording con condicionales

1. Scan category contains “tipo 2”

En NoSQL Workbench:

#	duration	notes	agent_name	configuration	chat	staff_id	category	timestamp	video_id	tags
1	5	[{"S": "Luis Suarez..."}, {"S": "05/11/2022..."}, {"S": "Este video es de prueba..."}]	Cristiano Ronaldo	[{"N": "0"}, {"N": "7"}, {"N": 720}, {"N": 1}]	[{"S": "Hola soy 3, en que te ayudo?,"}, {"S": "Quiero ponerle a prueba,"}]	2	[{"S": "tipo 2", "S": "tipo 3"}]	05/11/2022 10:02	https://www.youtube.com/	[{"S": "[prueba]"}]
2	2.2	[{"S": "Luis Suarez..."}, {"S": "05/11/2022..."}, {"S": "Este video es de prueba..."}]	Neymar junior	[{"N": "0"}, {"N": "3"}, {"N": 1080}, {"N": 1}]	[{"S": "Hola soy 4, en que te ayudo?,"}, {"S": "No sirve el internet,"}]	3	[{"S": "tipo 2", "S": "tipo 3", "S": "urgente"}]	05/11/2022 07:34	https://www.youtube.com/	[{"S": "[urgente]"}]

En Java (IntelliJ):

The screenshot shows the IntelliJ IDEA interface with the Java code for `AmazonDynamoDBScanGeneratedCodeRecordingCategoryTipo2.java`. The code is a scan operation on an Amazon DynamoDB table, filtering items where the `category` attribute is either 2 or 3. It uses AWS SDK for Java and includes annotations for `AmazonDynamoDBScanGeneratedCodeRecordingCategory` and `AmazonDynamoDBQueryGeneratedCode`. The run output window displays the log of the AWS Lambda function execution, showing the timestamp, log level (DEBUB), and the specific code being executed. The log includes details about the request ID, headers, and the connection to the AWS Lambda runtime.

```
23:02:07.683 [main] DEBUB org.apache.http.headers - http-outgoing-0 <> x-amz-crc32: 2442336107
23:02:07.683 [main] DEBUB org.apache.http.headers - http-outgoing-0 <> x-amzn-RequestID: 4b0e5c08-9a53-4304-be78-2285fc343500
23:02:07.688 [main] DEBUB org.apache.http.headers - http-outgoing-0 <> Content-length: 1121
23:02:07.689 [main] DEBUB org.apache.http.headers - http-outgoing-0 <> Server: Jetty(9.4.18.v20190429)
23:02:07.694 [main] DEBUB com.amazonaws.retry.ClockSkewAdjuster - Reported server date (from 'Date' header): Tue, 12 Apr 2022 04:02:00 GMT
23:02:07.719 [main] DEBUB org.apache.http.impl.execchain.MainClientExec - Connection can be kept alive for 60000 MILLISECONDS
23:02:07.719 [main] DEBUB org.apache.http.impl.DefaultHttpClientConnection - http-outgoing-0: set socket timeout to 0
23:02:07.722 [main] DEBUB org.apache.http.impl.com PoolingHttpClientConnectionManager - Connection released: [id: 0][route: {}->http://localhost:8000] can be kept alive for 60.0 seconds
23:02:07.722 [main] DEBUB org.apache.http.impl.com PoolingHttpClientConnectionManager - Connection released: [id: 0][route: {}->http://localhost:8000][total available: 1; route allocated: 1 of 50; total allocated: 1 of 50]
23:02:07.722 [main] DEBUB com.amazonaws.request - Received successful response: 200, AWS Request ID: 4b0e5c08-9a53-4304-be78-2285fc343500
23:02:07.722 [main] DEBUB com.amazonaws.requestId - x-amzn-RequestID: 4b0e5c08-9a53-4304-be78-2285fc343500
duration N[5]
notes L[[{"S: Luis Suarez,"}, {"S: 05/11/2022,"}, {"S: Este video es de prueba,"}]]
agent_name S[Christian Ronaldo]
configuration L[[{"N: 0,"}, {"N: 7,"}, {"N: 720,"}, {"N: 1,"}]]
chat L[[{"S: Hola soy 3, en que te ayudo?,"}, {"S: Quiero ponerte a prueba,"}]]
staff_id N[2]
category S$[[{"tipo 2, tipo 3"}]]
timestamp S[[05/11/2022 10:02]]
video_id S[[https://www.youtube.com/]]
tag S$[[prueba]]
customer S[[Maurin Lozano]]
successful_outcome 800L[false]
duration N[2]
notes L[[{"S: Luis Suarez,"}, {"S: 05/11/2022,"}, {"S: Una actuacion buena,"}]]
agent_name S[NEYMAR JUNIOR]
configuration L[[{"N: 0,"}, {"N: 3,"}, {"N: 480,"}, {"N: 1,"}]]
chat L[[{"S: Hola soy 4, en que te ayudo?,"}, {"S: Quiero comprar un perfume,"}]]
staff_id N[3]
category S$[[{"tipo 2, tipo 3, tipo 4"}]]
timestamp S[[05/11/2022 07:08]]
video_id S[[https://www.youtube.com/]]
tag S$[[complete, exito]]
customer S[[Tecatito Corona]]
successful_outcome 800L[false]
Scan successful.

Process finished with exit code 0
```

Entregables

1. Archivo .zip de IntelliJ
2. Archivo .json nuevo

Módulo 5. Calidad del Software

Timelog.

1. Registrar el timelog de la especificación de los requerimientos y del análisis del reto (entregables realizados en el bloque anterior), utilizando el script C16.

A continuación adjuntamos el archivo general del equipo:

[C16 Equipo](#)

2. De manera local e individual, iniciar con el registro del timelog de la fase de diseño para, posteriormente integrar los registros individuales en el timelog del equipo.

a) Entregar evidencia de un repositorio local en donde los integrantes del equipo, como parte de su estrategia de registro del Timelog, registren las actividades relevantes de la fase de diseño (fecha de inicio, fecha de terminación, tiempos de interrupción y su causa, tiempo real); para, posteriormente integrar el timelog de la fase de diseño.

A continuación adjuntamos los archivos individuales de cada integrante del equipo:

[Rebeca Monserrat Guevara Lagunas](#)

[Axel González Carreto](#)

[Jonatan Hernández García](#)

[Marcia Lechuga López](#)

Defectlog.

1. Registrar el defectlog de la especificación de los requerimientos y del análisis del reto, utilizando el script C14.

A continuación adjuntamos el archivo general del equipo:

C14 Equipo

2. De manera local e individual, iniciar con el registro del defectlog de la fase de diseño para, posteriormente integrar los registros individuales en el defectlog del equipo.

a) Entregar evidencia de un repositorio local en donde los integrantes del equipo, como parte de su estrategia de registro del Defectlog, registren los defectos de la fase de diseño (descripción del defecto y la fase en dónde se identificó, fase en la que fue injectado, tiempo dedicado a su corrección); para, posteriormente integrar el defectlog de la fase de diseño.

A continuación adjuntamos los archivos individuales de cada integrante del equipo:

Rebeca Monserrat Guevara Lagunas

Axel González Carreto

Jonatan Hernández García

Marcia Lechuga López

Posteriormente se enviará el link del repositorio de Github con las etapas/ciclos con base en los entregables solicitados

- Etapa0 o Ciclo0: Estrategia de aplicación del Timelog y Defectlog
- Etapa1 o Ciclo1: Nombre del entregable (puede ser: Timelog y Defctelog fase de requerimientos y análisis).