



Apellido y Nombre: Legajo:

Arquitectura (30)			Modelo de Dominio (40)				Persistencia (30)		Calificación final
Punto A (10)	Punto B (10)	Punto C (10)	Punto A (20)	Punto B (10)	Punto C (5)	Punto D (5)	Punto A (15)	Punto B (15)	

CartaGram

Contexto general

En esta oportunidad nos encargaremos de diseñar y desarrollar “CartaGram”, un servicio de mensajería online gratuito. Nos han solicitado que sean tres las formas posibles de utilizar este servicio: aplicación Móvil nativa, aplicación Web y aplicación Desktop.

Usuarios, grupos y mensajes

Cada usuario de CartaGram deberá tener una cuenta desde la cual podrá enviar y recibir mensajes hacia/de otros usuarios. Al ser enviados estos mensajes, en primer lugar, se deberán registrar en el servidor central y luego ser entregados al destinatario. Cada mensaje deberá contener un texto, una fecha/hora y uno o más adjuntos archivos.

Una vez que el usuario haya ingresado a su cuenta, cualquiera sea el medio, deberá poder enviar y recibir mensajes y ver todos los mensajes anteriores, sin importar el medio por el cual los haya enviado/recibido.

También se deberá brindar la posibilidad de crear grupos. Un grupo es un chat grupal entre varios usuarios, que llevan un nombre y una descripción que los identifica. Cuando un usuario envíe un mensaje a un grupo, el mismo deberá ser recibido por todos los participantes de éste.

Cada grupo deberá tener al menos un administrador, el cual tendrá la posibilidad de agregar o quitar usuarios, así como también cambiar el nombre y/o descripción del grupo y agregar o quitar otros administradores. Por defecto, el administrador inicial de un grupo será el usuario que creador del mismo.

Un usuario deberá poder ser parte de múltiples grupos y deberá poder salir de alguno de cualquiera cuando lo desee.

El Sistema deberá guardar trazabilidad sobre los mensajes ya que para este tipo de servicios es imprescindible saber si el mensaje fue recibido por el receptor y también si fue leído.

Bots

CartaGram deberá también otorgar la posibilidad de crear Bots. Los bots deberán funcionar de forma automatizada: cada vez que reciban un mensaje, deberán procesar el texto y determinar si el mismo es un comando. Un comando deberá poder ejecutar una o más acciones en función del mensaje que lo originó. Las acciones deberán poder ser personalizadas por los usuarios.

Por ejemplo:

- Chatbot:
 - Usuario “*Julietta*” hablando directamente con el chatbot
 - **Julietta:** set lenguaje español



- **chatbot:** ok!
- **Julieta:** traducir ingles “the cat is under the table”
- **chatbot:** “el gato está debajo de la mesa”
- **GativideoBot**
 - **Santino:** peliculas disponibles
 - **GativideoBot:** listado:
 - Mulan
 - Dune
 - ...
 - **Santino:** descargar “Dune”
 - **GativideoBot:** {link descarga pdf}
- **BotTruco**
 - En un grupo “*Amigos*” con amigo1, amigo2, amigo3 y amigo4 y BotTruco como integrantes
 - **amigo1:** jugar partida
 - **BotTruco** arma los equipos y manda un mensaje a cada integrante con las cartas que le tocó a cada uno
 - **amigo2** juega una carta 2
 - **BotTruco** le muestra a todos la carta que jugó
 - etc...

Se deberá tener en cuenta que las órdenes que se le podrán dar a los bots (comandos) deberán ser predeterminadas y en el caso de tener parámetros se los deberá poder extraer mediante una expresión regular. Por ejemplo:

traducir ingles “the cat is under the table” → matchea → traducir (idioma dst) “(text)”

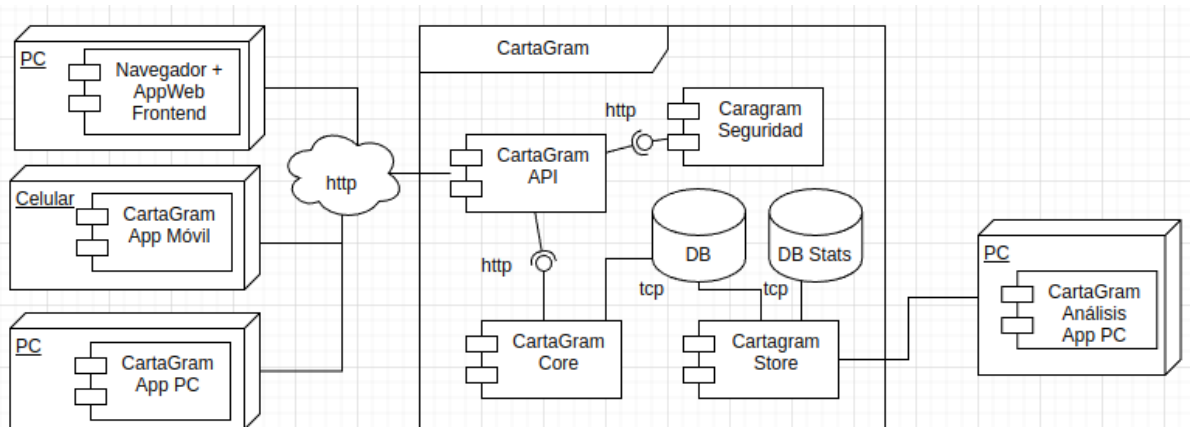
Alcance y requerimientos

El Sistema debe:

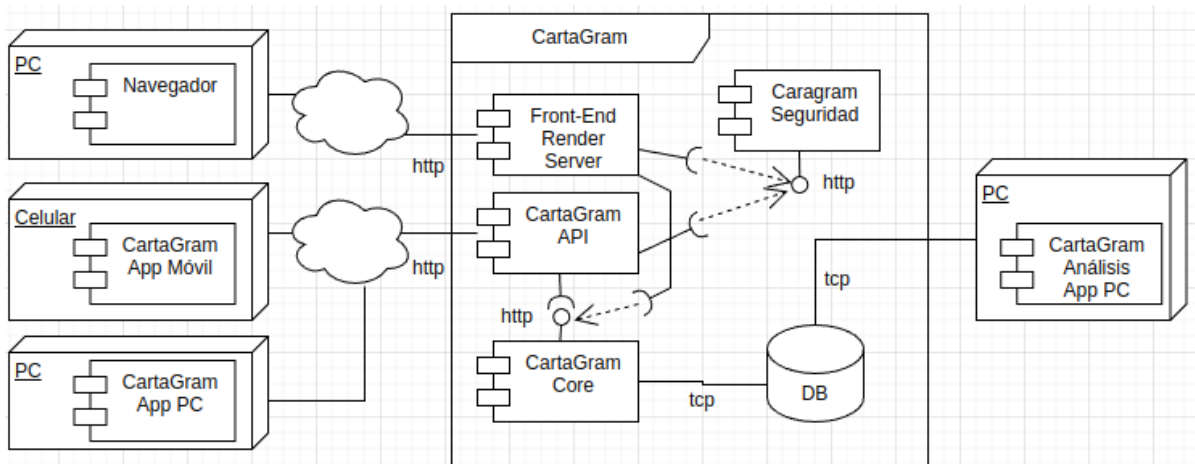
- Permitir el envío y recepción de mensajes entre usuarios.
- Guardar trazabilidad de los mensajes enviados/recibidos.
- Permitir la creación, modificación o eliminación de grupos.
- Permitir que los usuarios salgan de un grupo cuando lo deseen.
- Permitir la creación de bots, mediante la elección de comandos y personalización de acciones.

Punto 1 – Arquitectura (30 puntos)

Alternativa A



Alternativa B



- CartaGram Core es nuestro componente, el que implementa las reglas de negocio del contexto. Dentro de él se ejecutan los bots.
 - CartaGram API es la API REST que es apuntada por componentes que se ejecutan en clientes externos.
 - CartaGram Seguridad y CartaGram Core exponen una API REST para comunicarse con los otros componentes que se ejecutan en la infraestructura de la empresa.
 - FrontEnd Render Server es una aplicación web cuyo motor de templates se ejecuta en el servidor.
 - CartaGram Seguridad autentica y autoriza mensajes, verifica políticas, recupera contraseñas, etc..
 - CartaGram Store toma los datos de operatoria diaria (DB) y los coloca en una base NoSQL documental para agilizar los análisis de datos
 - CartaGram Análisis es una App utilizada por los miembros de la empresa para analizar los datos de uso y generar estadísticas, modelos, etc...
1. **(10 puntos)** Especifique los métodos que considere más importantes del componente “CartaGram API”. Utilice el nivel de detalle que considere adecuado y haga explícitas **todas** sus suposiciones.
 2. **(10 puntos)** Explique las diferencias en la facilidad de mantenimiento de las alternativas A y B.



3. **(10 puntos)** Dado el siguiente escenario, indicar el atributo de calidad que considere que está implicado en el problema detallado, definirlo y comparar cómo impactaría la elección de las alternativas A o B.

Estímulo	CartaGram Análisis realiza una consulta que procesa un gran volumen de datos
Ambiente	Producción
Respuesta	El servicio de mensajería no sufre ninguna degradación en su nivel de servicio

Punto 2 - Modelo de Dominio (40 puntos)

- A. **(20 Puntos)** Documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
- B. **(10 Puntos)** Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. También puede optar por justificar mediante código, pseudocódigo o algún otro diagrama complementario.
- C. **(5 puntos)** Explicar cómo se agrega un nuevo Bot. Mencione un ejemplo
- D. **(5 puntos)** Explicar cómo resuelve el siguiente escenario: *“El Usuario X le pregunta a GativideoBot el listado de películas disponibles”*. Puede utilizar código, pseudo-código u otros diagramas (diagrama de secuencia, por ejemplo).

Punto 3 – Persistencia (30 puntos)

- A. **(20 Puntos)** Diseñar el modelo de datos del punto anterior para poder persistir en una base de datos relacional, indicando las entidades con sus respectivos campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda.
- B. **(10 Puntos)** Justificar:
- Qué elementos del modelo es necesario persistir.
 - Cómo resolvió los *impedance mismatches*.
 - Las estructuras de datos que deban ser desnormalizadas, si corresponde.

NOTA: Explicar supuestos y justificar decisiones de diseño.

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.