

# Chat Adivinador

# Planteamiento

- Crea un chat “dummy” que te conteste las preguntas que hagas y vaya mostrando la conversación en un ListView

# Layout

4G

12:23

Guesschat

¿Deberia tomar un tequila?

No

¿Que tal un vodka?

Es muy probable

¿Agua?

Si

:|

ENVIAR

# Posibles Respuestas

- Si
- No
- Pregunta de nuevo
- Es muy probable
- No lo creo

# Objetivos Principales

- Crea un chat con respuestas aleatorias y muéstralo en un ListView

# Tip Importante

- Cuando agregas o editas datos en un adapter, debes informarle acerca de los cambios escribiendo `myAdapter.notifyDataSetChanged();`

# Objetivos Secundarios

- Asigna un fondo blanco a los elementos de pregunta y un fondo de otro color para las respuestas, puedes tomar colores geniales de <https://flatuicolors.com/>
- Asigna un gravity start para las preguntas y un gravity end para las respuestas (Observa la imagen como las preguntas están del lado izquierdo y las respuestas del lado derecho)

# Objetivos Secundarios

- Valida que exista una pregunta y muestra un Toast si el usuario no insertó una y da click en el botón enviar
- Limpia el campo de preguntas cuando se envíe una.



# Objetivo Pro

- Haz que la lista haga scroll automáticamente para ir viendo las últimas respuestas. Tip: puedes ver al respecto aquí <http://stackoverflow.com/questions/3606530/listview-scroll-to-the-end-of-the-list-after-updating-the-list>

# Tips (¡Spoiler Alert!)

- Puedes usar un objeto con un campo de tipo boolean que te diga si el texto a mostrar en el ListView es pregunta o respuesta para insertar el background.
- Genera un entero aleatorio y usa un switch para decidir qué respuesta dar
- Utiliza el método add() para añadir ítems a la lista

# Tips (¡Spoiler Alert!)

- Puedes guardar colores en la carpeta `res > values > colors` y después tomarlos usando `ContextCompat(context, R.color.mi_color)`
- Android trae sus propios colores comunes incluidos, por ejemplo, puedes tomar el color blanco con `android.R.color.white`