



# **TECNOLOGICO DE ESTUDIOS SUPERIORES DE ECATEPEC**

## **DIVISION DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

### **INTEGRANTES:**

Cuevas Marín Mario Alberto

Garduño Rangel Bárbara Poulette

Morales Martínez César

Palafox Rodríguez Eduardo Daniel

Solís Pérez Christian André

Torres Sánchez Edgar Yael

### **GRUPO:**

**5701**

### **MATERIA:**

Gestión de proyectos Software

### **PARCIAL:**

3er Parcial

## Contenido

Costos y tiempo.....	3
Organización interna .....	4
WBS .....	5
Matriz de Riesgo.....	5
Pruebas de cada actividad.....	7
Acceso a la tienda en Top.gg.....	7
Programación del reproductor de colas.....	8
Comando clear .....	8
Comando disconnect.....	9
Comando grab .....	10
Comando help .....	12
Comando invite .....	14
Comando loop .....	15
Comando loopqueue.....	16
Comando np .....	18
Comando pause.....	20
Comando play.....	22
Comando queue .....	27
Comando resume .....	30
Comando skip.....	32
Comando skipto .....	33
Comando stats.....	35
Perfil de Recursos.....	39
Cronograma (Porcentajes, Costos, Seguimiento) .....	46
Trello .....	47

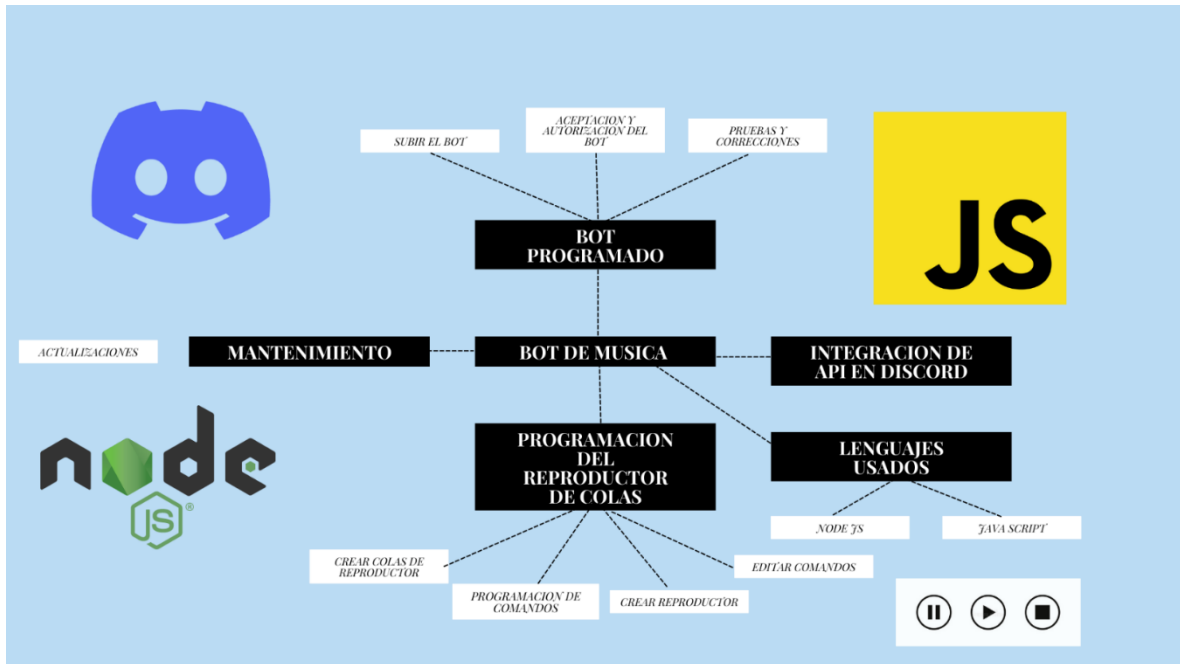
Costos y tiempo

DETALLE DE DESARROLLO DE PROYECTO	
7 PERSONAS	
TIEMPO: 6 Meses	
COSTO CLIENTE: \$ 500,500.00	
COSTO PROYECTO: \$ 385,000.00	
GANANCIA ESTIMADA: \$ 115,000.00	
PORCENTAJE ESTIMADO: 30 %	

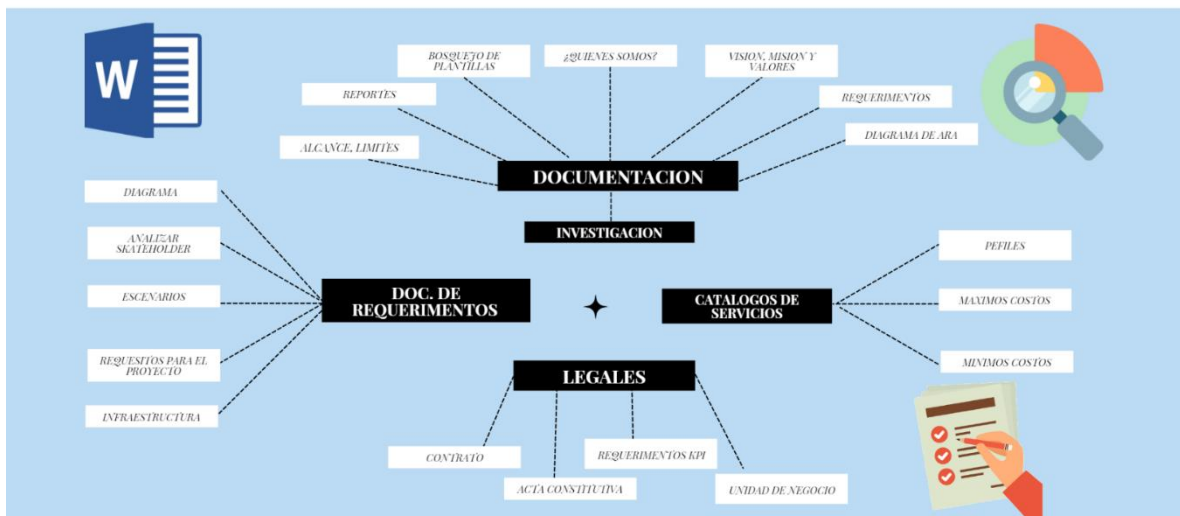
ROLES DE PERSONAS	COSTO POR PERSONA X MES	TIEMPO MESES	TOTAL
Jefe de proyecto	\$20,000.00	6	\$120,000.00
Asesor de Software	\$8,000.00	2	\$16,000.00
Analista de Software	\$14,000.00	2	\$28,000.00
Diseñador de Software	\$15,000.00	3	\$45,000.00
Desarrollador de Software	\$30,000.00	4	\$120,000.00
Tester	\$16,000.00	2	\$32,000.00
Desarrollador Web	\$12,000.00	2	\$24,000.00

TOTAL GLOBAL DEL PROYECTO	\$385,000.00
---------------------------	--------------

## Organización interna



## BOT DE MUSICA PARA DISCORD



## WBS



## Matriz de Riesgo

- **Publicidad de YouTube**

Existe la posibilidad de una problemática con YouTube ya que el software no implementará los anuncios de la plataforma, que, aunque solo transmita el audio es factible un posible reclamo por parte de YouTube

- **Renuncia de empleados**

Un factor del cual podría generar algunas problemáticas con el desarrollo del software en dado caso de que se presenten

- **Problemas financieros**

Esto puede ocurrir en dado caso de que se gaste más dinero del presupuesto previsto para el desarrollo del software

- **Pandemias/catástrofes naturales**

Situaciones externas que impedirían el desarrollo del mismo software

- **Prestadores de servicios**

Problemáticas con deficiencias con servicios como podría ser la luz

- **Leyes/Impedimentos legales**

Factor que implique una limitación o impedimento en el desarrollo del software por algún posible incumplimiento de algún factor legal

- **Incumplimiento**

En situaciones donde él o los contratistas no efectúen la parte del contrato previamente firmada

- **Robo de información**

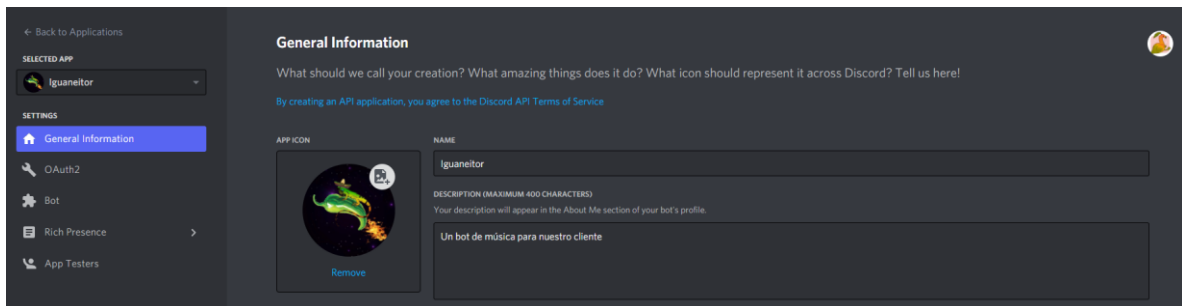
Situación que implicaría que un tercero robe información del desarrolló del software en cuestión

		Consecuencia				
		Insignificante	Menor	Moderado	Mayor	Catastrófico
Probabilidad	Constante (5)		Renuncia de empleados			
	Moderado (4)			Prestadores de servicios		
	Ocasional (3)			Publicidad de YouTube	Pandemias catástrofes	
	Posible (2)		Leyes Impedimentos		Problemas financieros	Robo de información
	Improbable (1)			Incumplimiento		

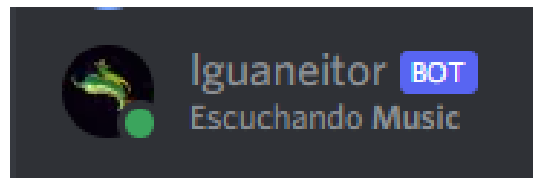
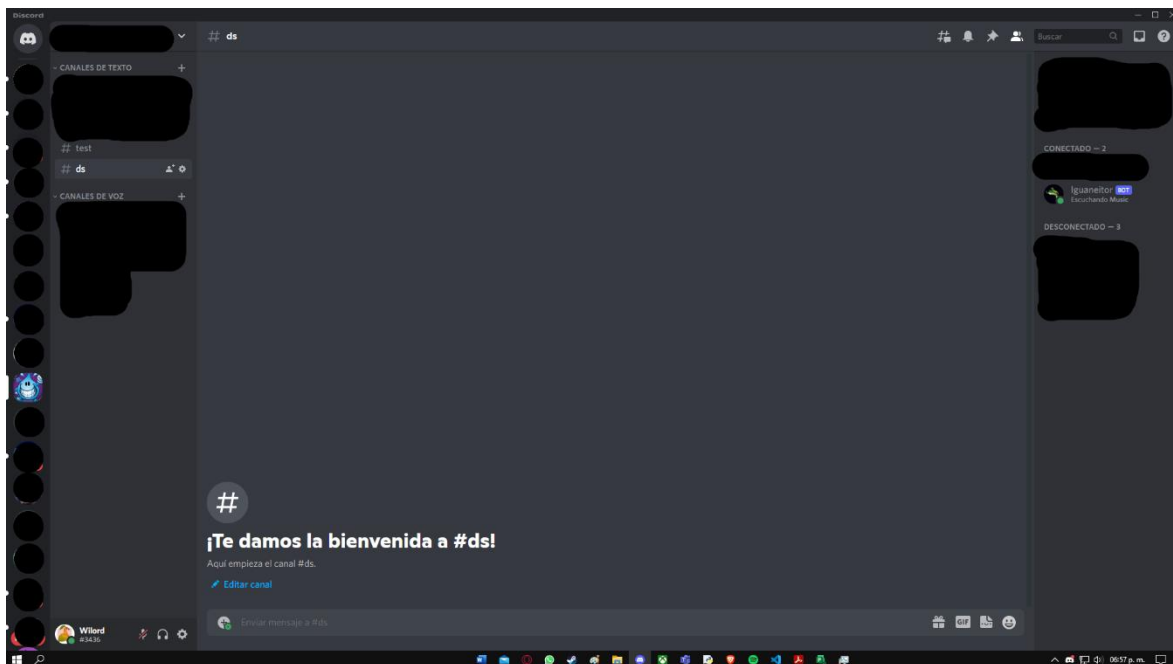
## Pruebas de cada actividad

Acceso a la tienda en Top.gg

Darse de alta en el programa de Desarrolladores de Discord



## Integración de la API en Discord



## Investigación del lenguaje

[Link al Documento de Investigación](#)

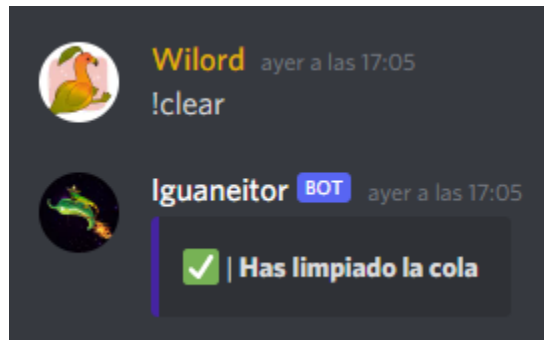
## Programación del reproductor de colas

Desarrollo de la lógica del Bot para los comandos

### Comando clear/limpiar

```
3  module.exports = {
4    name: "clear",
5    description: "Borra la cola actual",
6    usage: "",
7    permissions: {
8      channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
9      member: [],
10   },
11   aliases: ["cl", "cls"],
12   /**
13    * @param {import("../structures/DiscordMusicBot")} client
14    * @param {import("discord.js").Message} message
15    * @param {string[]} args
16    * @param {*} param3
17    */
18   run: async (client, message, args, { GuildDB }) => {
19     let player = await client.Manager.get(message.guild.id);
20     if (!player)
21       return client.sendTime(
22         message.channel,
23         "❌ | **No se está reproduciendo nada...**"
24       );
25
26     if (!player.queue || !player.queue.length || player.queue.length === 0)
27       return client.sendTime(
28         message.channel,
29         "❌ | **No hay una lista...**"
30       );
31     if (!message.member.voice.channel)
32       return client.sendTime(
33         message.channel,
34         "❌ | Debes estar en un canal de voz para usar este comando."
35       );
36     if (
37       message.guild.me.voice.channel &&
38       message.member.voice.channel.id !== message.guild.me.voice.channel.id
39     )
40       return client.sendTime(
41         message.channel,
42         "❌ | **¡Debemos estar en el mismo chat de voz primero!**"
43       );
44     player.queue.clear();
45     await client.sendTime(message.channel, "✅ | **Has limpiado la cola**");
46   },
```





Comando disconnect/deconectar

```
module.exports = {
  name: "disconnect",
  description: "Desconecta al bot del chat",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["dc"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    await client.sendTime(message.channel, ":notes: | **Desconectado**");
    await message.react("✅");
    player.destroy();
  },
};
```



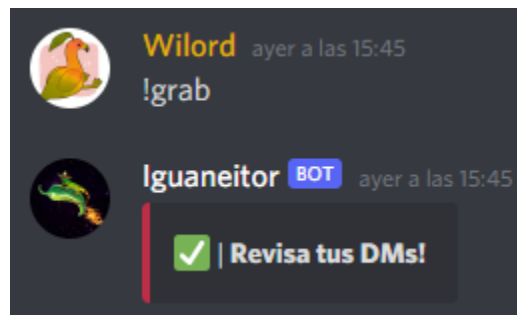
Comando grab/guardar

```
module.exports = {
  name: "grab",
  description: "¿Te gustó la canción? Te la enviamos por DM",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["save"],
  /** ...
  >
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
    >     return client.sendTime(...
    >     );
    if (!player.playing)
    >     return client.sendTime(...
    >     );
    if (!message.member.voice.channel)
    >     return client.sendTime(...
    >     );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
    >     return client.sendTime(...
    >     );
    message.author
    >     .send(
    >       new MessageEmbed()
    >       .setAuthor(
    >         `Canción guardada`,
    >         client.user.displayAvatarURL({
    >           dynamic: true,
    >         })
    >       )
    >       .setThumbnail(
    >         `https://img.youtube.com/vi/${player.queue.current.identifier}/mqdefault.jpg`
    >       )
    >       .setURL(player.queue.current.uri)
    >       .setColor(client.botconfig.EmbedColor)
    >       .setTitle(`**${player.queue.current.title}**`)
    >       .addField(
    >         `🕒 Duración: `,
    >         `${prettyMilliseconds(player.queue.current.duration, {
    >           colonNotation: true,
```

```

    `⌚ Duración: `,
    ```${prettyMilliseconds(player.queue.current.duration, {
      colonNotation: true,
    })}``,
    true
  )
  .addField(`🎵 Autor: `, `${player.queue.current.author}`, true)
  .addField(
    `▶ Reproducir:`,
    `${
      GuildDB ? GuildDB.prefix : client.botconfig.DefaultPrefix
    }play ${player.queue.current.uri}`
  )
  .addField(`💬 Guardado en:`, `<#${message.channel.id}>`)
  .setFooter(
    `Solicitado por: ${player.queue.current.requester.tag}`,
    player.queue.current.requester.displayAvatarURL({
      dynamic: true,
    })
  )
  )
  .catch((e) => {
    return message.channel.send("**:x: Tus mensajes directos están desactivados**");
  });
  client.sendTime(message.channel, "✅ | **Revisa tus DMs!**");
},

```



Comando help/ayuda

```
module.exports = {
  name: "help",
  description: "Información sobre todo... casi",
  usage: "[comando]",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["h"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let Commands = client.commands.map(
      (cmd) =>
        `${GuildDB ? GuildDB.prefix : client.botconfig.DefaultPrefix}${
          cmd.name
        }${cmd.usage ? " " + cmd.usage : ""}\\` - ${cmd.description}`
    );

    let Embed = new MessageEmbed()
      .setAuthor(
        `Comandos de ${client.user.username}`,
        client.botconfig.IconURL
      )
      .setColor(client.botconfig.EmbedColor)
      .setFooter(
        `Para obtener info de cada comando escribe ${
          GuildDB ? GuildDB.prefix : client.botconfig.DefaultPrefix
        }help [comando] | Tenga un buen día`
      )
      .setDescription(`${Commands.join("\n")}`);

    Iguaneitor Version: v1.0
    [:sparkles: Página Web](`${
      client.botconfig.SupportServer
    }`);
    if (!args[0]) message.channel.send(Embed);
    else { ...
  }
}
```



**Wilord** hoy a las 23:00

!help



**Iguaneitor** BOT hoy a las 23:00



### Comandos de Iguaneitor

**!clear** - Borra la cola actual  
**!disconnect** - Desconecta al bot del chat  
**!grab** - ¿Te gustó la canción? Te la envíamos por DM  
**!help [comando]** - Información sobre todo... casi  
**!invite** - Invítame a tu servidor  
**!loop** - Repite la canción actual  
**!loopqueue** - Repite la lista completa  
**!nowplaying** - Mira qué canción se está reproduciendo  
**!pause** - Pausa la música  
**!play [canción]** - Reproduce tu canción favorita  
**!queue** - Muestra todas las canciones que estén en cola  
**!resume** - Reproduce la canción desde donde la pausaste  
**!search [canción]** - Muestra resultados de canciones de acuerdo a lo solicitado  
**!skip** - Salta la canción actual  
**!skipto <núm en cola>** - Salta hacia una canción de la cola  
**!stats** - Obtén el estado del bot

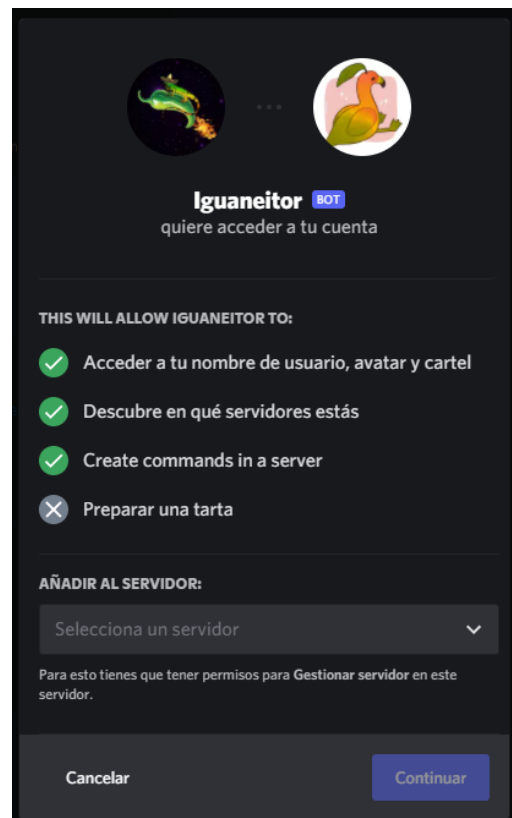
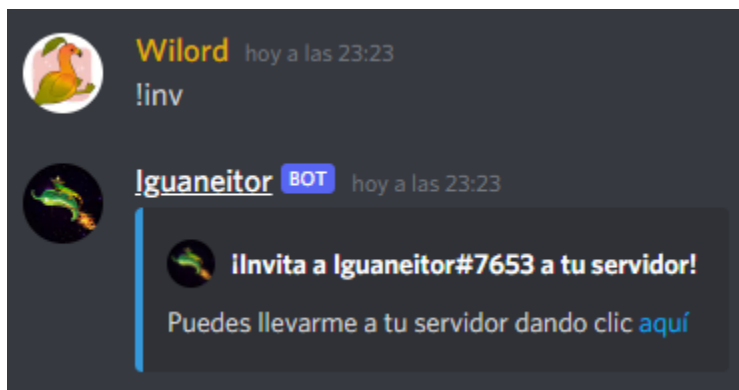
Iguaneitor Version: v1.0

🌟 [Página Web](#)

Para obtener info de cada comando escribe !help [comando] | Tenga un buen día!

## Comando invite/invitar

```
module.exports = {
  name: "invite",
  description: "Invítame a tu servidor",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["inv"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let embed = new MessageEmbed()
      .setAuthor(
        "¡Invita a " + client.user.tag + " a tu servidor!",
        client.user.displayAvatarURL()
      )
      .setColor("BLUE")
      .setDescription(
        `Puedes llevarme a tu servidor dando clic [aquí](https://discord.com/oauth2/authorize?client_id=${
          client.botconfig.ClientID
        }&permissions=${
          client.botconfig.Permissions
        }&scope=bot%20${client.botconfig.Scopes.join("%20")}&redirect_url=${
          client.botconfig.Website
        }&${client.botconfig.CallbackURL}&response_type=code)`
      );
    message.channel.send(embed);
  },
};
```



## Comando loop/repetir

```
module.exports = {
  name: "loop",
  description: "Repite la canción actual",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["l", "lt"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando.**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | **Debemos estar en el mismo chat de voz primero!**"
      );

    if (player.trackRepeat) {
      player.setTrackRepeat(false);
      client.sendTime(message.channel, `🔊 \`Desactivado\``);
    } else {
      player.setTrackRepeat(true);
      client.sendTime(message.channel, `🔊 \`Activado\``);
    }
  },
};
```

## Comando loopqueue/repetirlista

```
module.exports = {
  name: "loopqueue",
  description: "Repite la lista completa",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["lq"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        ":x: | **;Debemos estar en el mismo canal de voz primero!**"
      );
    if (player.queueRepeat) {
      player.setQueueRepeat(false);
      client.sendTime(message.channel, `repeat: Cola en loop \`Activado\``);
    } else {
      player.setQueueRepeat(true);
      client.sendTime(message.channel, `repeat: Cola en loop \`Desactivado\``);
    }
  },
};
```





Comando nowplaying/reproduciendoahora

```
module.exports = {
  name: "nowplaying",
  description: "Mira qué canción se está reproduciendo",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["np"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );

    let song = player.queue.current;
    let QueueEmbed = new MessageEmbed()
      .setAuthor("Reproduciendo", client.botconfig.IconURL)
      .setColor(client.botconfig.EmbedColor)
      .setDescription(`${song.title} (${song.uri})`)
      .addField("Solicitado por", `${song.requester}`, true)
      .addField(
        "Duración",
        `${
          client.ProgressBar(player.position, player.queue.current.duration, 15)
            .Bar
        } \`${prettyMilliseconds(player.position, {
          colonNotation: true,
        })} / \`${prettyMilliseconds(player.queue.current.duration, {
          colonNotation: true,
        })}\``
      )
      .setThumbnail(player.queue.current.displayThumbnail());
    return message.channel.send(QueueEmbed);
  },
};
```



**Wilord** hoy a las 23:30

!np



**Iguaneitor** **BOT** hoy a las 23:30



**Reproduciendo**

Sneaky Driver

**Solicitado por**

@Wilord

**Duración**

0:00 / 4:40



## Comando pause/pausar

```
module.exports = {
  name: "pause",
  description: "Pausa la música",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: [],
  /**
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | **Debemos estar en el mismo chat de voz primero!**"
      );
    if (player.paused)
      return client.sendTime(
        message.channel,
        "❌ | **La música ya está en pausa!**"
      );
    player.pause(true);
    let embed = new MessageEmbed()
      .setAuthor(`Pausado`, client.botconfig.IconURL)
      .setColor(client.botconfig.EmbedColor)
      .setDescription(`Escribe \`${GuildDB.prefix}resume\` para seguir reproduciendo!`);
    await message.channel.send(embed);
    await message.react("✅");
  },
};
```



**Wilord** hoy a las 23:31

!pause



1



**Iguaneitor** **BOT** hoy a las 23:31



**Pausado**

Escribe **!resume** para seguir reproduciendo!

## Comando play/reproducir

```
module.exports = {
  name: "play",
  description: "Reproduce tu canción favorita",
  usage: "[canción]",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["p"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | ❗;Debes estar en un chat de voz primero!"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | ❗;Debemos estar en el mismo chat de voz!"
      );
    let SearchString = args.join(" ");
    if (!SearchString)
      return client.sendTime(
        message.channel,
        `❗Usage - ❗\`${GuildDB.prefix}play [song]\``
      );
    let CheckNode = client.Manager.nodes.get(client.botconfig.Lavalink.id);
    let Searching = await message.channel.send(":mag_right: Buscando...");
    if (!CheckNode || !CheckNode.connected) {
      return client.sendTime(
        message.channel,
        "❌ | ❗Lavalink no conectado❗"
      );
    }
  }
}
```

```

const player = client.Manager.create({
  guild: message.guild.id,
  voiceChannel: message.member.voice.channel.id,
  textChannel: message.channel.id,
  selfDeafen: client.botconfig.ServerDeafen,
  volume: client.botconfig.DefaultVolume,
});

let SongAddedEmbed = new MessageEmbed().setColor(
  client.botconfig.EmbedColor
);

if (!player)
  return client.sendTime(
    message.channel,
    "❌ | **No se está reproduciendo nada...**"
  );

if (player.state !== "CONNECTED") await player.connect();

try {
  if (SearchString.match(client.Lavasfy.spotifyPattern)) {
    await client.Lavasfy.requestToken();
    let node = client.Lavasfy.nodes.get(client.botconfig.Lavalink.id);
    let Searched = await node.load(SearchString);

    if (Searched.loadType === "PLAYLIST_LOADED") {
      let songs = [];
      for (let i = 0; i < Searched.tracks.length; i++)
        songs.push(TrackUtils.build(Searched.tracks[i], message.author));
      player.queue.add(songs);
      if (
        !player.playing &&
        !player.paused &&
        player.queue.totalSize === Searched.tracks.length
      )
        player.play();
      SongAddedEmbed.setAuthor(
        `Playlist añadida a la cola`,
        message.author.displayAvatarURL()
      );
      SongAddedEmbed.addField(
        "Tienes",
        `\\${Searched.tracks.length}\\` en cola`,
        false
      );
    }
  }
}

```

```

Searching.edit(SongAddedEmbed);
} else if (Searched.loadType.startsWith("TRACK")) {
    player.queue.add(
        TrackUtils.build(Searched.tracks[0], message.author)
    );
    if (!player.playing && !player.paused && !player.queue.size)
        player.play();
    SongAddedEmbed.setAuthor(`Añadidos a la cola`, client.botconfig.IconURL);
    SongAddedEmbed.setDescription(
        `[${Searched.tracks[0].info.title}](${Searched.tracks[0].info.uri})`
    );
    SongAddedEmbed.addField(
        "Autor",
        Searched.tracks[0].info.author,
        true
    );
    if (player.queue.totalSize > 1)
        SongAddedEmbed.addField(
            "Posición en cola",
            `${player.queue.size - 0}`,
            true
        );
    Searching.edit(SongAddedEmbed);
} else {
    return client.sendTime(
        message.channel,
        "***No se encontraron coincidencias para - ***" + SearchString
    );
}
else {
    let Searched = await player.search(SearchString, message.author);
    if (!player)
        return client.sendTime(
            message.channel,
            "❌ | ***No se está reproduciendo nada...***"
        );

    if (Searched.loadType === "NO_MATCHES")
        return client.sendTime(
            message.channel,
            "***No se encontraron coincidencias para - ***" + SearchString
        );
}

```



```

    }
    else if (Searched.loadType == "PLAYLIST_LOADED") {
        player.queue.add(Searched.tracks);
        if (
            !player.playing &&
            !player.paused &&
            player.queue.totalSize === Searched.tracks.length
        )
            player.play();
        SongAddedEmbed.setAuthor(
            `Playlist añadida a la cola`,
            client.botconfig.IconURL
        );
        SongAddedEmbed.setDescription(
            `${Searched.playlist.name}(${Searched.tracks.length})`
        );
        SongAddedEmbed.addField(
            "Tienes ",
            `${Searched.tracks.length} en cola`,
            false
        );
        SongAddedEmbed.addField(
            "Duración de la Playlist",
            `${prettyMilliseconds(Searched.playlist.duration, {
                colonNotation: true,
            })}`,
            false
        );
        Searching.edit(SongAddedEmbed);
    } else {
        player.queue.add(Searched.tracks[0]);
        if (!player.playing && !player.paused && !player.queue.size)
            player.play();
        SongAddedEmbed.setAuthor(`Añadido a la cola`, client.botconfig.IconURL);
        SongAddedEmbed.setDescription(
            `${Searched.tracks[0].title}(${Searched.tracks[0].uri})`
        );
        SongAddedEmbed.addField("Autor", Searched.tracks[0].author, true);
        SongAddedEmbed.addField(
            "Duración",
            `${prettyMilliseconds(Searched.tracks[0].duration, {
                colonNotation: true,
            })}`,
            true
        );
    }
}

```

```

    );
    if (player.queue.totalSize > 1)
        SongAddedEmbed.addField(
            "Posición en la cola",
            `${player.queue.size - 0}`,
            true
        );
    Searching.edit(SongAddedEmbed);
}
} catch (e) {
    console.log(e);
    return client.sendTime(
        message.channel,
        "***No se encontraron coincidencias para - ***" + SearchString
    );
}
},

```



**Wilord** ayer a las 18:58

!play <https://www.youtube.com/watch?v=TFS316SzGFQ&t=57s>

YouTube

**Carlos AGlez**

[Himno de la URSS \(Гимн Советского Союза\). Traducción al español.](#)



**Iguaneitor** BOT ayer a las 18:58

Buscando... (editado)



**Añadido a la cola**

[Himno de la URSS \(Гимн Советского Союза\). Traducción al español.](#)

**Autor**

Carlos AGlez

**Duración**

3:45



**Reproduciendo ahora**

[Himno de la URSS \(Гимн Советского Союза\). Traducción al español.](#)

**Solicitada por**

@Wilord

**Duración**

3:45



Comando queue/cola

```
module.exports = {
  name: "queue",
  description: "Muestra todas las canciones que estén en cola",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["q"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );

    if (!player.queue || !player.queue.length || player.queue === 0) {
      let QueueEmbed = new MessageEmbed()
        .setAuthor("Reproduciendo ahora", client.botconfig.IconURL)
        .setColor(client.botconfig.EmbedColor)
        .setDescription(
          `[${player.queue.current.title}](${player.queue.current.uri})`
        )
        .addField("Solicitado por", `${player.queue.current.requester}`, true)
        .addField(
          "Duración",
          `${
            client.ProgressBar(
              player.position,
              player.queue.current.duration,
              15
            ).Bar
          } \`${prettyMilliseconds(player.position, {
            colonNotation: true,
          })} / ${prettyMilliseconds(player.queue.current.duration, {
            colonNotation: true,
          })}\``
        )
    }
  }
}
```

```

        .setThumbnail(player.queue.current.displayThumbnail());
        return message.channel.send(QueueEmbed);
    }

    let Songs = player.queue.map((t, index) => {
        t.index = index;
        return t;
    });

    let ChunkedSongs = _.chunk(Songs, 10); //How many songs to show per-page

    let Pages = ChunkedSongs.map((Tracks) => {
        let SongsDescription = Tracks.map(
            (t) =>
                `\\${t.index + 1}.\\` [`${t.title}]`(`${t.uri})` \\n\\` ${prettyMilliseconds(
                    t.duration,
                    {
                        colonNotation: true,
                    }
                )} \\`
            )
        ).join("\\n");

        let Embed = new MessageEmbed()
            .setAuthor("Cola", client.botconfig.IconURL)
            .setColor(client.botconfig.EmbedColor)
            .setDescription(
                `**Reproduciendo ahora:** \\n[${player.queue.current.title}]`(`${player.queue.current.uri})` \\n\\n**Siguiente:** \\n${SongsDescription}\\n\\n`
            )
            .addField("Canciones en total: \\n", `\\` ${player.queue.totalSize - 1}\\``, true)
            .addField(
                "Duración total: \\n",
                `\\` ${prettyMilliseconds(player.queue.duration, {
                    colonNotation: true,
                })} \\``,
                true
            )
            .addField("Requested by:", `${player.queue.current.requester}`, true)
            .addField(
                "Duración de la canción actual:",
                `${
                    client.ProgressBar(
                        player.position,
                        player.queue.current.duration,
                        15
                    ).Bar
                }`
            );

        return Embed;
    });

    if (!Pages.length || Pages.length === 1)
        return message.channel.send(Pages[0]);
    else client.Pagination(message, Pages);
},

```

```

        ).Bar
    } \\` ${prettyMilliseconds(player.position, {
        colonNotation: true,
    })} / \\` ${prettyMilliseconds(player.queue.current.duration, {
        colonNotation: true,
    })} \\`
    )
    .setThumbnail(player.queue.current.displayThumbnail());

    return Embed;
});

if (!Pages.length || Pages.length === 1)
    return message.channel.send(Pages[0]);
else client.Pagination(message, Pages);
},

```



Iguaneitor BOT 08/11/2021



## Queue

### Currently Playing:

Delusive Bunker



### Up Next:

1. Full Confession

4:06 | Requested by: @Wilord

2. You Will Never Know

3:16 | Requested by: @Wilord

3. Beach House - Space Song [LYRIC VIDEO Spanish / English]

Subtitulado Español

5:21 | Requested by: @Wilord

4. FKJ & Masego - Tadow

8:07 | Requested by: @Wilord

5. Chinatown

6:03 | Requested by: @Wilord

6. In My Head

5:55 | Requested by: @Wilord

7. M.O.O.N. - 'Crystals' [Hotline Miami Soundtrack]

4:50 | Requested by: @Wilord

8. M.O.O.N. - 'Hydrogen' [Hotline Miami Soundtrack]

4:51 | Requested by: @Wilord

9. iamthekidyouknowwhatimean - Run

4:54 | Requested by: @Wilord

10. All for Now

3:25 | Requested by: @Wilord

Total songs:

17

Total length:

2:25:08

Requested by:

@Wilord

Current song duration:

0:19.8 / 5:18



Pág 1/2

Comando resume/reanudar

```
module.exports = {
  name: "resume",
  description: "Reproduce la canción desde donde la pausaste",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: [],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | **Debemos estar en el mismo chat de voz primero!**"
      );
    if (player.playing)
      return client.sendTime(
        message.channel,
        "❌ | **La música ya se está reproduciendo!**"
      );
    player.pause(false);
    await message.react("✅");
  },
};
```



**Wilord** hoy a las 23:43

!resume



**Iguaneitor** **BOT** hoy a las 23:43

❌ | La música ya se está reproduciendo!



**Wilord** hoy a las 23:43

!pause



1



**Iguaneitor** **BOT** hoy a las 23:43

⏸ **Pausado**

Escribe `!resume` para seguir reproduciendo!



**Wilord** hoy a las 23:43

!resume



1

Comando skip/saltar

```
module.exports = {
  name: "skip",
  description: "Salta la canción actual",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["s", "next"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    let player = await client.Manager.get(message.guild.id);
    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | **¡Debemos estar en el mismo chat de voz primero!**"
      );
    player.stop();
    await message.react("✅");
  },
};
```





Comando skipto/saltarhacia

```
module.exports = {
  name: "skipto",
  description: `Salta hacia una canción de la cola`,
  usage: "<núm en cola>",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["st"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message, args, { GuildDB }) => {
    const player = client.Manager.create({
      guild: message.guild.id,
      voiceChannel: message.member.voice.channel.id,
      textChannel: message.channel.id,
      selfDeafen: client.botconfig.ServerDeafen,
    });

    if (!player)
      return client.sendTime(
        message.channel,
        "❌ | **No se está reproduciendo nada...**"
      );
    if (!message.member.voice.channel)
      return client.sendTime(
        message.channel,
        "❌ | **Debes estar en un canal de voz para usar este comando**"
      );
    if (
      message.guild.me.voice.channel &&
      message.member.voice.channel.id !== message.guild.me.voice.channel.id
    )
      return client.sendTime(
        message.channel,
        "❌ | **¡Debemos estar en el mismo chat de voz primero!**"
      );
  }
};
```

```

try {
  if (!args[0])
    return client.sendTime(
      message.channel,
      `**Sintaxis**: \`${GuildDB.prefix}skipto [número]\``
    );
  //if the wished track is bigger then the Queue Size
  if (Number(args[0]) > player.queue.size)
    return client.sendTime(
      message.channel,
      `❌ | Esta canción no está en la cola. Prueba de nuevo.`
    );
  //remove all tracks to the jumped song
  player.queue.remove(0, Number(args[0]) - 1);
  //stop the player
  player.stop();
  //Send Success Message
  return client.sendTime(
    message.channel,
    `🔊 Te has saltado \`${Number(args[0]) - 1}\` canciones`
  );
} catch (e) {
  console.log(String(e.stack).bgRed);
  client.sendError(message.channel, "Algo salió mal.");
}
},
};
};

```



**Wilord** hoy a las 23:50  
!skipto 2



**Iguaneitor** BOT hoy a las 23:50

🔊 Te has saltado 1 canciones



**Reproduciendo ahora** 🎵

Nier Automata Soundtrack - 03 Peaceful Sleep

**Solicitada por**

@Wilord

**Duración**

6:51



## Comando stats/estado

```
module.exports = {
  name: "stats",
  description: "Obtén el estado del bot",
  usage: "",
  permissions: {
    channel: ["VIEW_CHANNEL", "SEND_MESSAGES", "EMBED_LINKS"],
    member: [],
  },
  aliases: ["about", "ping", "info"],
  /**
   *
   * @param {import("../structures/DiscordMusicBot")} client
   * @param {import("discord.js").Message} message
   * @param {string[]} args
   * @param {*} param3
   */
  run: async (client, message) => {
    const { version } = require("discord.js");
    cpuStat.usagePercent(async function (err, percent, seconds) {
      if (err) {
        return console.log(err);
      }
    });
    const duration = moment
      .duration(message.client.uptime)
      .format(" D[d], H[h], m[m]");

    const embed = new MessageEmbed();
    embed.setColor(client.botconfig.EmbedColor);
    embed.setTitle(`Estado de \`${client.user.username}\``);
    embed.addFields(
      [
        {
          name: ":ping_pong: Ping",
          value: `~\`${Math.round(client.ws.ping)}ms\``,
          inline: true,
        },
        {
          name: ":clock1: Tiempo activo",
          value: `~\`${duration}\``,
          inline: true,
        },
        {
          name: ":file_cabinet: Memoria",
          value: `~\`${(process.memoryUsage().heapUsed / 1024 / 1024).toFixed(
            2
          )} GB\``
        }
      ]
    );
    message.channel.send({ embed });
  }
};
```

```

embed.addFields(
  {
    name: ":homes: Servers",
    value: `🏠 ${client.guilds.cache.size}`,
    inline: true,
  },
  {
    name: ":busts_in_silhouette: Usuarios",
    value: `👤 ${client.users.cache.size}`,
    inline: true,
  },
  {
    name: ":control_knobs: API Latencia",
    value: `🔧 ${message.client.ws.ping}ms`,
    inline: true,
  }
);
embed.addFields(
  {
    name: ":robot: Versión",
    value: `🤖 1.0`,
    inline: true,
  },
  {
    name: ":blue_book: Discord.js",
    value: `📘 v${version}`,
    inline: true,
  },
  {
    name: ":green_book: Node",
    value: `📖 ${process.version}`,
    inline: true,
  }
);
return message.channel.send(embed);
});
},

```



 **Wilord** hoy a las 23:29  
!stats



 **Iguaneitor** BOT hoy a las 23:29


**Estado de Iguaneitor**

 <b>Ping</b> 📶 62ms	 <b>Tiempo activo</b> 🕒 13m	 <b>Memoria</b> 💾 22.53mb
 <b>Servers</b> 🏠 1	 <b>Usuarios</b> 👤 7	 <b>API Latencia</b> 🔧 62ms
 <b>Versión</b> 🤖 v1.0	 <b>Discord.js</b> 📘 v12.5.3	 <b>Node</b> 📖 v16.13.0

Crear/Implementar Reproductor

**Iguaneitor**

**Wilord**


**Añadido a la cola**  
**La Cucaracha**


Autor	Duración	Posición en la cola
Serena Rossi Fernández	3:05	23


Diseñar logo



## Crear colas de reproducción

**Iguaneitor BOT** hoy a las 23:46

**Cola**





**Reproduciendo ahora:**  
**Breath of a Serpent**




**Siguiente:**

1. **Little Dark Age**  
5:00 | Solicitado por: **@Wilord**
2. **Delusive Bunker**  
5:18 | Solicitado por: **@Wilord**
3. **Full Confession**  
4:06 | Solicitado por: **@Wilord**
4. **You Will Never Know**  
3:16 | Solicitado por: **@Wilord**
5. **Beach House - Space Song [LYRIC VIDEO Spanish / English]**  
Subtitulado Español  
5:21 | Solicitado por: **@Wilord**
6. **FKJ & Masego - Tadow**  
8:07 | Solicitado por: **@Wilord**
7. **Chinatown**  
6:03 | Solicitado por: **@Wilord**
8. **In My Head**  
5:55 | Solicitado por: **@Wilord**
9. **M.O.O.N. - 'Crystals' [Hotline Miami Soundtrack]**  
4:50 | Solicitado por: **@Wilord**
10. **M.O.O.N. - 'Hydrogen' [Hotline Miami Soundtrack]**  
4:51 | Solicitado por: **@Wilord**

**Canciones en total:** 19      **Duración total:** 1:36:37      **Requested by:** **@Wilord**

**Duración de la canción actual:**  
 3:14.9 / 3:37

 Pág 1/2

 1     1     1

## Pruebas

[Link al video](#)

## Perfil de Recursos

**Puesto: Project Manager**

**Clave: 001**

### **Descripción:**

Un Project manager o jefe de proyecto es el directivo de un área de la compañía que se encarga de definir la estrategia para realizar un proyecto concreto y supervisar su ejecución. Gracias a su capacidad de trabajo en equipo y de coordinación de áreas, podrá seguir el desarrollo de las distintas etapas de un proyecto, anticipar problemas, redistribuir los recursos físicos y humanos para su mejor eficiencia, todo ello con una meta: que el proyecto llegue a buen puerto, en tiempo y forma.

### **Funciones:**

- Facilitar la planificación del o los proyectos que se le asignen.
- Realizar análisis financieros.
- Modificar y asegurar la documentación de procesos.
- Validar y revisar el tema de excelencia operativa.
- Desarrollar modelos de mejora continua, mediante la evaluación y rediseño de estrategias

### **Perfil:**

- Ser optimista y positivo.
- Ser colaborativo.
- Manejar relaciones y conflictos.
- Comunicación.
- Ser respetuoso, cortés con el cliente y compañeros de trabajo.
- Mostrar integridad.
- Dar crédito a otros cuando lo merecen.
- Centrarse en las cosas importantes.
- Visión holística y sistemática del proyecto.
- Aplicación del pensamiento crítico.

### **Relación con otro puesto:**

Entra directamente en relación con todos los puestos (Asesor de software, Analista de Software, Diseñador de Software, Desarrollador de Software, Tester y Desarrollador web) ya que será quien les pante el proyecto para poder trabajar.

**Puesto: Asesor de software**

**Clave: 002**

**Descripción:**

Su función principal es la de asesorar al cliente en cuanto a la implementación de tecnologías y herramientas necesarias para el desarrollo de su negocio.

**Funciones:**

- Seguimiento, cotizaciones y prospección
- Visitar a los clientes para promocionar los productos y servicios
- Asesoría técnica para la generación y cierre de proyectos.
- Hay que asegurar que el producto llegue en tiempo y forma al cliente
- Seguimiento a facturación y cobranza
- Generación de alianzas estratégicas para procesos comercial

**Perfil:**

- Escucha Activa y empatía por clientes. Capacidad para identificar y evaluar inquietudes de clientes.
- Tolerancia y saber mantener la calma con los clientes.
- Trabajo en equipo por objetivos Iniciativa por auto aprendizaje. Flexibilidad para constantes cambios de ritmo de trabajo.

**Relación con otro puesto:**

Entra directamente en relación con el Project Manager para poder asesorar bien al cliente sobre su proyecto.



**Puesto: Analista de software**

**Clave: 003**

**Descripción:**

Los analistas de software adaptan y diseñan sistemas de información para ayudar al diseñador de software trabajar de forma más rápida y eficiente.

**Funciones:**

- Participación y moderación en sesiones de levantamiento de requerimientos de software
- Evaluación de documentos de requerimientos para identificar puntos de mejora y comunicación con dueños de proceso
- Escritura de documentos de requerimientos de software
- Modelado de procesos para comunicación de alcance
- Definición de pruebas de usuario y validaciones de requerimientos
- Liderar y negociar el alcance y definición de productos

**Perfil:**

- Comprensión de lectura
- Investigación
- Escritura
- Análisis y modelación
- Liderazgo en moderación y discusión de requerimientos
- Observación
- Organización
- Relaciones interpersonales y empatía
- Creatividad
- Habilidades para entrevistar

**Relación con otro puesto:**

Entra directamente en relación con el Diseñador de Software, Desarrollador de software y Desarrollador Web.

**Puesto: Diseñador de Software**

**Clave: 004**

**Descripción:**

Diseño de software es el proceso de diseño para la planificación de una solución de software. Este proceso es, por regla general, necesario para que los programadores puedan manejar la complejidad que la mayoría de los programas informáticos poseen y para disminuir el riesgo de desarrollos erróneos.

**Funciones:**

- Crear aplicaciones WEB Front-end
- Manejo de IDE´s (Visual estudio, Visual code y Pycharm)
- Diseño y programación: C#, Java, Java Script y Python
- Patrones de Diseño MVC
- Diseño de sitios WEB y maquetado de aplicaciones
- Dominio de Photoshop, illustrator, InDesign y XD
- Manejo de Versiones IDE's (Visual estudio, Visual code y Pycharm)
- Llevar Control de versiones

**Perfil:**

- Conocimientos en desarrollo de Software
- Diseño y documentación de Manuales en sus diferentes etapas y folletos
- Conocimiento de Bases de Datos Relacionales y no relacionales
- Experiencia en Web Services SOAP y REST API, MVC y .Net
- Conocimiento Programación orientada a objetos y/o funcional

**Relación con otro puesto:**

Entra directamente en relación con el Analista de software, Desarrollador de software y Desarrollador Web.

**Puesto: Desarrollador de software**

**Clave: 005**

**Descripción:**

Se encargan de desarrollar los pilares de los sistemas operativos que son creados por los Programadores y de probar el código de nuevos programas para garantizar su eficiencia. Asimismo, realizan pruebas de calidad en nuevos proyectos antes de su lanzamiento.

**Funciones:**

- Desarrollar soluciones centradas en los clientes dentro de un equipo multidisciplinario conformado por asesores empresariales, otros desarrolladores, y el cliente.
- Tener control sobre los módulos/aplicaciones del software asignados y brindar soluciones de software de calidad y en tiempo para los proyectos.
- Diseñar/developar soluciones efectivas y flexibles dentro de Odoo de aplicaciones de terceros integrables.
- Solucionar, eliminar fallos, arreglar y actualizar el software, así como asegurar que las soluciones de software funcionen firmemente en el campo
- Dar soporte a cualquier migración o actualización de actividades en el software.
- Adherirse y promover mejores prácticas de programación como normas de codificación, codificar, diseñar, examinar y poner a prueba la cobertura.

**Perfil:**

- Frontend: 1+ año de experiencia trabajando con Typescript/Javascript
- Fronted: conocimiento funcional de HTML y SASS/LESS/CSS
- Conocimiento de arquitectura de bases de datos y diseño de conceptos (UML/ERD)
- Habilidad para eliminar fallos y optimizar soluciones personalizadas existentes.
- Habilidad para comunicar conceptos técnicos de forma efectiva con los miembros del equipo, socios comerciales y clientes.
- Plataforma Unix/Linux

**Relación con otro puesto:**

Entra directamente en relación con el Analista de software, Diseñador de Software y Teaster.

**Puesto: Tester**

**Clave: 006**

**Descripción:**

Los probadores de software (también conocidos como testers, su denominación en inglés) planifican y llevan a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente. Identifican el riesgo de sufrir errores de un software, detectan errores y los comunican. Evalúan el funcionamiento general del software y sugieren formas de mejorarlo.

**Funciones:**

- Realizar los diferentes tipos de prueba en las aplicaciones empresariales.
- Proponer mejoras a los sistemas de acuerdo a las reglas del negocio.
- Comunicación amplia con el equipo de trabajo para lograr resultados.

**Perfil:**

- Proactivo
- Eficiente
- Responsable
- Trabajo en equipo
- Atención a los detalles
- Concentración
- Manejo de SQL Server realizando consultas básicas.
- Nociones de programación, uso postman, lectura de JSON
- Pruebas Unitarias
- Pruebas de Componentes
- Pruebas de Integración

**Relación con otro puesto:**

Entra directamente en relación con el Diseñador de Software y Desarrollador de software.

**Puesto: Desarrollador web Jr**

**Clave: 007**

**Descripción:**

Un Desarrollador Junior o Junior Developer trabaja en un equipo de desarrollo escribiendo y manteniendo código para aplicaciones informáticas. Es un desarrollador de software de nivel básico que ayuda al equipo de desarrollo con todos los aspectos de diseño y codificación de software.

**Funciones:**

**Mantenimiento ERP.**

- Mantenimiento y diseño de bases de datos.
- Acceso a servidores con aplicaciones ERP y configuración de servidores.
- Análisis de sistema y documentación de código.
- Control de versiones.
- APIs

**Perfil:**

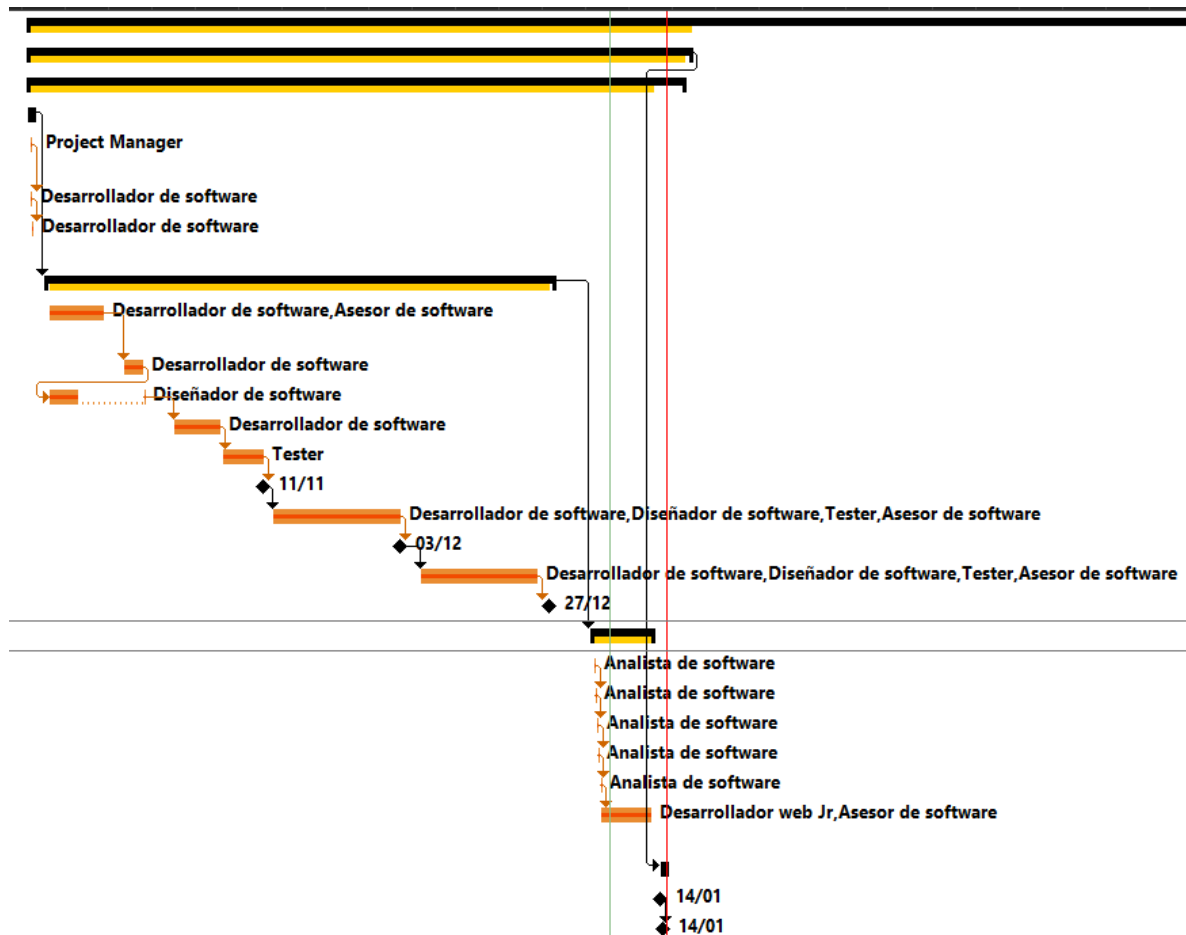
- Conocimientos avanzados en lenguaje JavaScript.
- Desarrollo de Aplicaciones en Angular.
- Manejo de PHP avanzado, LINUX SERVER, GITHUB, NODE JS, SQL, MYSQL, HTML.
- Conocimiento en documentación de código.

**Relación con otro puesto:**

Entra directamente en relación con el Analista de software y Diseñador de Software.

## Cronograma (Porcentajes, Costos, Seguimiento)

1 BOT DE MUSICA	Prorratio	141,975.00	\$141,975.00	141,975.00	\$0.00	99%	lun 04/10/21	vie 01/07/22	197 días
1.1 Desarrollo	Prorratio	141,975.00	\$141,975.00	141,975.00	\$0.00	100%	lun 04/10/21	mar 18/01/22	79 días
1.1.1 Creación de Bot	Prorratio	141,975.00	\$141,975.00	141,975.00	\$0.00	100%	lun 04/10/21	lun 17/01/22	78 días
1.1.1.1 Acceso a la tienda en top.gg	Prorratio	\$1,062.50	\$1,062.50	\$1,062.50	\$0.00	100%	lun 04/10/21	lun 04/10/21	8 horas
1.1.1.1.1 Darse de alta en el programa de desarrolladores en Discord	Prorratio	\$125.00	\$125.00	\$125.00	\$0.00	100%	lun 04/10/21	lun 04/10/21	1 hora
1.1.1.1.2 Integración de API discord	Prorratio	\$375.00	\$375.00	\$375.00	\$0.00	100%	lun 04/10/21	lun 04/10/21	2 horas
1.1.1.1.3 Investigación del lenguaje NodeJS/JavaScript	Prorratio	\$562.50	\$562.50	\$562.50	\$0.00	100%	lun 04/10/21	lun 04/10/21	3 horas
1.1.1.2 Programación del reproductor y col	Prorratio	134,125.00	\$134,125.00	134,125.00	\$0.00	100%	jue 07/10/21	lun 27/12/21	59 días
1.1.1.2.1 Desarrollo de la lógica del Bot para los comandos	Prorratio	\$10,125.00	\$10,125.00	\$10,125.00	\$0.00	100%	jue 07/10/21	vie 15/10/21	6.75 días
1.1.1.2.2 Crear reproductor	Prorratio	\$5,250.00	\$5,250.00	\$5,250.00	\$0.00	100%	mar 19/10/21	vie 22/10/21	3.5 días
1.1.1.2.3 Diseño de un logo	Prorratio	\$2,250.00	\$2,250.00	\$2,250.00	\$0.00	100%	jue 07/10/21	vie 22/10/21	3 días
1.1.1.2.4 Crear colas de Reproduccion	Prorratio	\$9,000.00	\$9,000.00	\$9,000.00	\$0.00	100%	mié 27/10/21	mié 03/11/21	6 días
1.1.1.2.5 Pruebas	Prorratio	\$4,000.00	\$4,000.00	\$4,000.00	\$0.00	100%	jue 04/11/21	mié 10/11/21	5 días
1.1.1.2.6 Entrega Primer Avance	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	100%	jue 11/11/21	jue 11/11/21	0 días
1.1.1.2.7 Tiempo para correcciones	Prorratio	\$51,750.00	\$51,750.00	\$51,750.00	\$0.00	100%	vie 12/11/21	jue 02/12/21	15 días
1.1.1.2.8 Entrega Segundo Avance	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	100%	vie 03/12/21	vie 03/12/21	0 días
1.1.1.2.9 Tiempo para correcciones	Prorratio	\$51,750.00	\$51,750.00	\$51,750.00	\$0.00	100%	lun 06/12/21	vie 24/12/21	15 días
1.1.1.2.10 Finalización del desarrollo del	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	100%	lun 27/12/21	lun 27/12/21	0 días
1.1.1.3 Documentación del bot	Prorratio	\$6,787.50	\$6,787.50	\$6,787.50	\$0.00	100%	lun 03/01/22	mié 12/01/22	7.88 días
1.1.1.3.1 Como conectar el bot a discord	Prorratio	\$175.00	\$175.00	\$175.00	\$0.00	100%	lun 03/01/22	lun 03/01/22	2 horas
1.1.1.3.2 Código de reproductores	Prorratio	\$175.00	\$175.00	\$175.00	\$0.00	100%	lun 03/01/22	lun 03/01/22	2 horas
1.1.1.3.3 Comandos	Prorratio	\$262.50	\$262.50	\$262.50	\$0.00	100%	lun 03/01/22	lun 03/01/22	3 horas
1.1.1.3.4 Editar comandos	Prorratio	\$87.50	\$87.50	\$87.50	\$0.00	100%	lun 03/01/22	lun 03/01/22	1 hora
1.1.1.3.5 Reproducir musica y programar	Prorratio	\$87.50	\$87.50	\$87.50	\$0.00	100%	mar 04/01/22	mar 04/01/22	1 hora
1.1.1.3.6 Documentación completa en una página web	Prorratio	\$6,000.00	\$6,000.00	\$6,000.00	\$0.00	100%	mar 04/01/22	mié 12/01/22	6 días
1.2 Entrega final	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	0%	vie 14/01/22	vie 14/01/22	0.75 días
1.2.1 Página de documentacion	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	0%	vie 14/01/22	vie 14/01/22	0 días
1.2.2 Entrega del bot	Prorratio	\$0.00	\$0.00	\$0.00	\$0.00	0%	vie 14/01/22	vie 14/01/22	0 días



# Trello

000 Tablero **iguaneitor** Bot Música Visible para el Espacio de trabajo ES CM B C E +1 Invitar

**Lista de tareas**

Entregar segundo avance

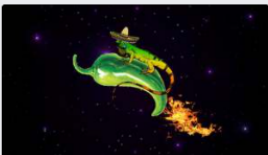
3 de dic.

B CM C ES E M

Segundo periodo de correcciones

6 de dic. - 24 de dic.

B CM C ES E M



Entrega Final

14 de ene. de 2022 1

B CM C ES E M

+ Añada una tarjeta

**En proceso**

Documentación

3 de ene. de 2022 - 11 de ene. de 2022

B CM C ES M

primer periodo de correcciones

12 de nov. - 2 de dic.

B CM C ES E M

+ Añada una tarjeta

**Hecho**

Entregar primer avance

11 de nov.

B CM C ES E M

Propuesta

ES

Pruebas(Testeo de funcionamiento)

4 de nov. - 10 de nov.

B CM C ES E M

Desarrollo de los comandos

7 de oct. - 19 de oct.

CM ES

Casos de uso

CM

Análisis de puesto

B

Contrato

M

Acta constitutiva

+ Añada una tarjeta

**Minutas Realizadas**

Minuta: 1

4 de oct.

Minuta 2

5 de oct.

Minuta 3

6 de oct.

Minuta 4

7 de oct.

Minuta 5

18 de oct.

Minuta 6

19 de oct.

Minuta 7

27 de oct.

Minuta 8

2 de nov.

+ Añada una tarjeta

//trello.com/c/yvOrZSa2/10-entregar-segundo-avance

## Conclusiones

Cuevas Marín Mario Alberto

En este proyecto aprendimos a trabajar con tiempos, presupuestos, presión y en equipo ya que todo está muy pero muy cuadrado ya que todo se tenía que entregar en tiempo y forma además de que también checamos para que funcionasen todos los demás aprendizajes, el Bot es de mucha ayuda para los que utilizan Discord es excelente aprender todo esto para un futuro y para el trabajo.

Morales Martínez César

En este proyecto aprendimos a trabajar con tiempos, presupuestos, presión y en equipo ya que todo está muy pero muy cuadrado ya que todo se tenía que entregar en tiempo y forma además de que también checamos para que funcionasen todos los demás aprendizajes, lo que si no fue de ayuda fue la guía durante el proyecto ya que el cliente nunca respeto las fechas o las cambiaba sin previo aviso, nos ayudó a entender que clientes no querer.

Garduño Rangel Bárbara Poulette

El proyecto fue terminado de manera correcta y satisfactoria, nuestro problema más persistente fue el uso del Project ya que tuvimos que prender a utilizarlo de forma independiente también aprendimos a cómo hacer un presupuesto correcto de un proyecto de software así como la investigación de los sueldos, tiempos, servidores y todo lo que conlleva el desarrollo de un software.

Solís Pérez Christian André

Podemos mencionar que el proyecto fue concluido correctamente, aunque en su debido momento hubo ciertas dificultades con la conexión en Lavalink, hubo un buen nivel de organización en general para la distribución de tareas y pendientes, aprendimos a cómo gestionar y emplear mejor Project y al tener cierta experiencia del cómo se desarrollaría un equipo de trabajo para un proyecto como empresa.

Torres Sánchez Edgar Yael

El proyecto ha pasado por momentos duros y también uno que otro contratiempo, pero a pesar de todo se ha podido mantener a flote y para sorpresa de algunos ha sido finalizado. Esto empezó más como una forma de explorar nuevas áreas y terminó como un proyecto personal que me gustaría seguir desarrollando.

Palafox Rodríguez Eduardo Daniel

Se ha hecho realidad una gran idea para acabar un poco con la piratería y a la vez ofrecer un inmenso catálogo musical, con esta aplicación ofrece a sus usuarios millones de canciones para que puedan ser escuchadas mediante el uso de la aplicación en Discord y por lo tanto, mientras no necesites moverla a otros soportes ayuda a resolver las soluciones y sea más fácil de escuchar música por medio de Discord.