

Documentación Taco Paco

Santi Martínez

13 de noviembre de 2025

lstsetlenguaje=Javascript

Índice

1. Introducción	3
2. Arquitectura de la app	4
3. APP Móvil	7
4. APP Escritorio	10
5. API Rest	11
5.1. Flujo de la API con las APPs	11
5.2. Componentes	12
5.2.1. Models/	13
6. Conclusión	14
7. Bibliografía	15

1. Introducción

Taco Paco es una aplicación de comida rápida en la que el cliente puede reservar su mesa, y a partir de ahí seleccionar sus productos favoritos.

La app de escritorio forma parte únicamente del establecimiento, ya que su funcionalidad radica en cobrar pedidos y liberar las mesas. Gracias a ello podrá ver también las mesas libres y las ocupadas.

Ambas se conectan entre sí a través de una API Rest elaborada con express y mongoose, el último para guardar y editar los cambios en Mongo Atlas, la base de datos de la aplicación.

2. Arquitectura de la app

Diagrama de flujo

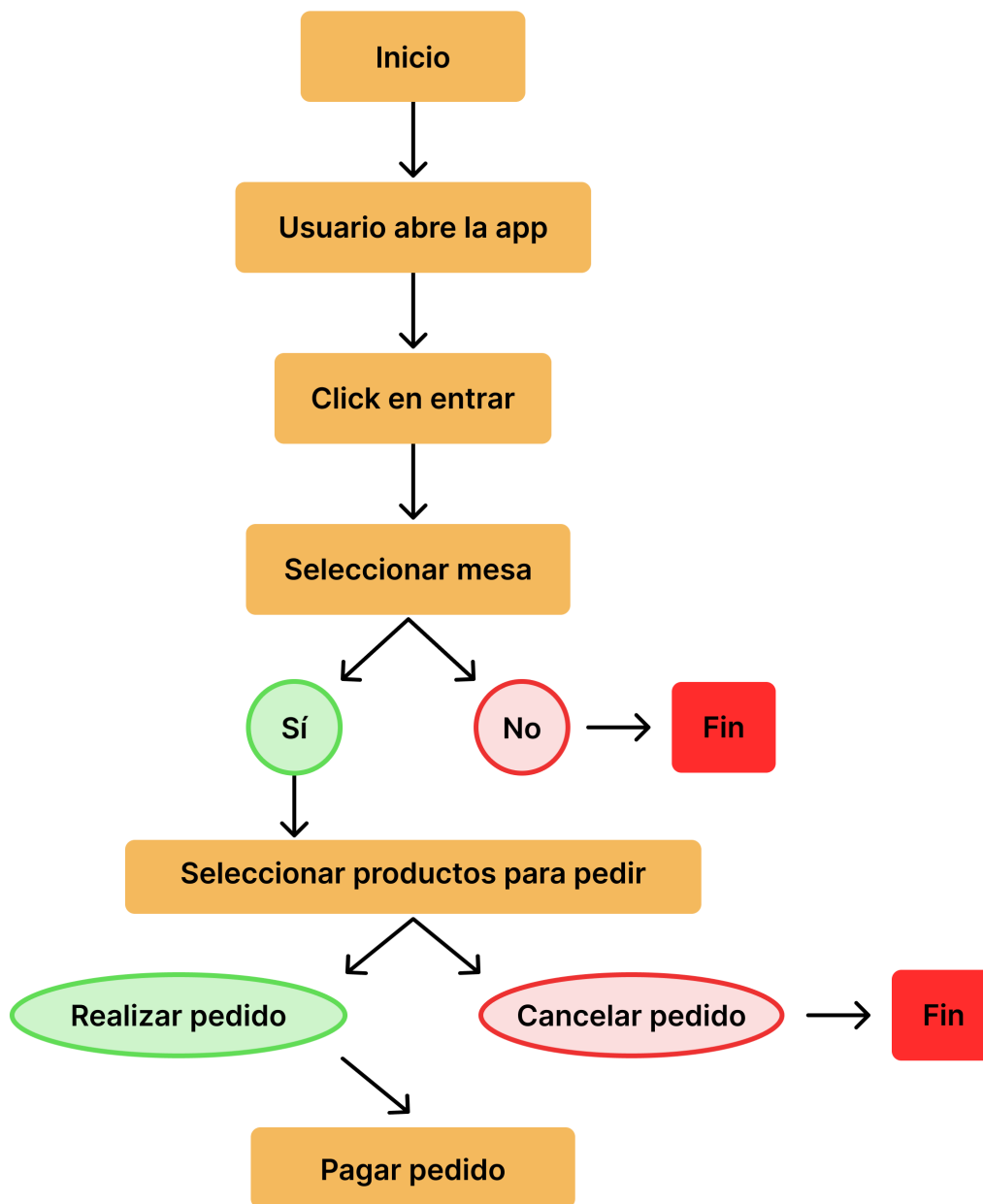


Figura 1: Diagrama de flujo

Diagrama de grafo

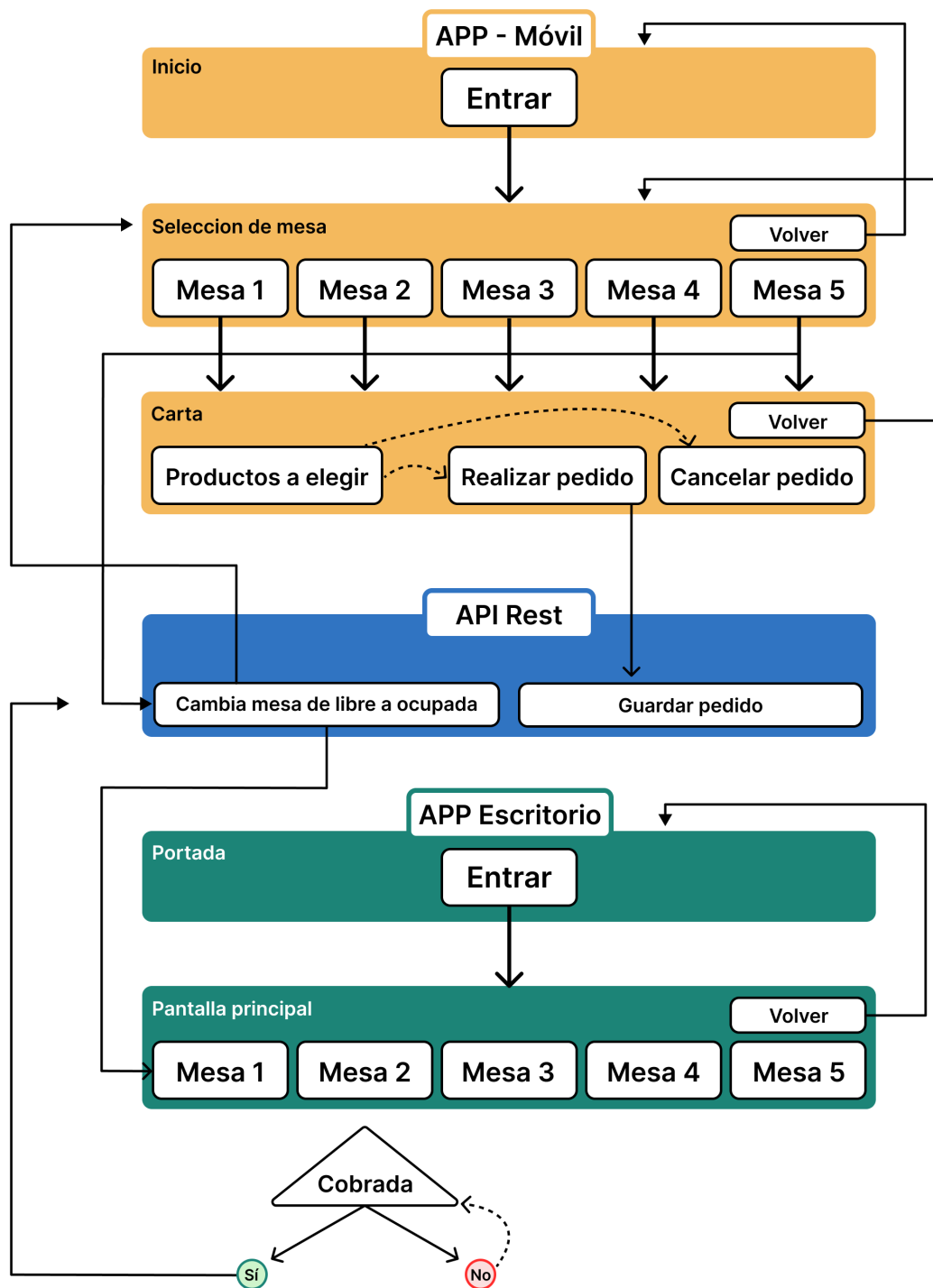


Figura 2: Diagrama de grafo

Diagrama de casos de uso

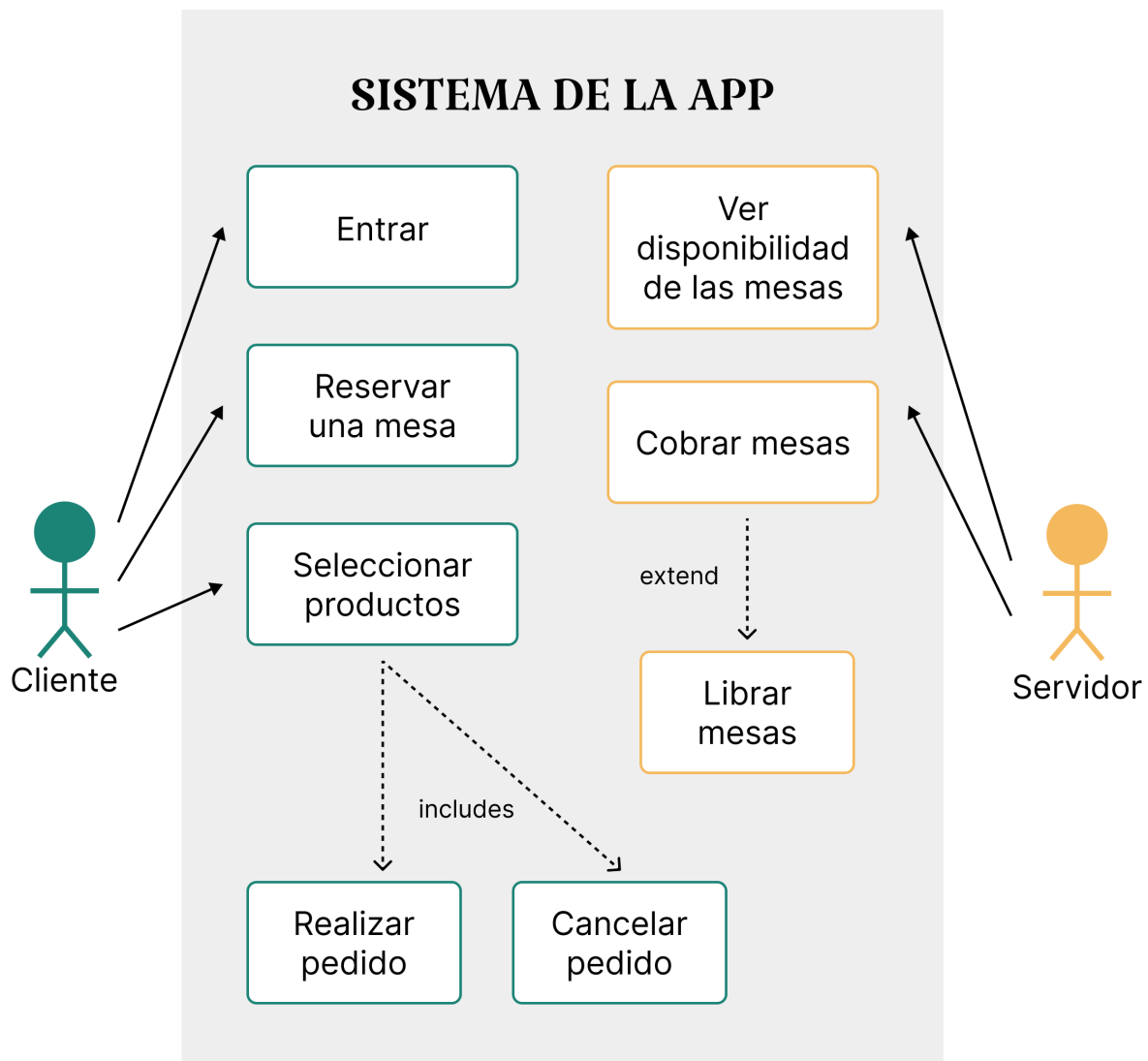


Figura 3: Diagrama de casos de uso

3. APP Móvil

Estructura de la app

- app/
 - sampledata/
 - manifest/
 - AndroidManifest
 - kotlin+java/
 - activities/
 - ◇ Carta
 - ◇ EleccionMesa
 - ◇ MainActivity
 - clases/
 - ◇ Mesa
 - ◇ Pedido
 - servicios/
 - ◇ Api
 - ◇ RetrofitClient
 - java (generated)/
 - resources/
 - drawable/
 - layout/
 - ◇ carta
 - ◇ eleccion mesa
 - ◇ inicio
 - mipmap/
 - values/
 - xml/
 - resources (generated)/
- Gradle Scripts/

Imágenes de la app

Pantalla principal

Primera pantalla que se encuentra el usuario al iniciar la app, la única funcionalidad que tiene es la que le da el botón **Entrar**, el cual como su nombre indica, sirve para entrar en el siguiente menú de la app.

En la pantalla se encuentra el nombre de la empresa junto a su año en la que se fundó. También el botón de entrar que es el encargado de cambiar entre activities.



CASO 1: Todas las mesas libres

Esta es la pantalla en la que se muestran las mesas disponibles del establecimiento. Se trata de un menú con 5 mesas:

- 3 interiores
- 2 terraza

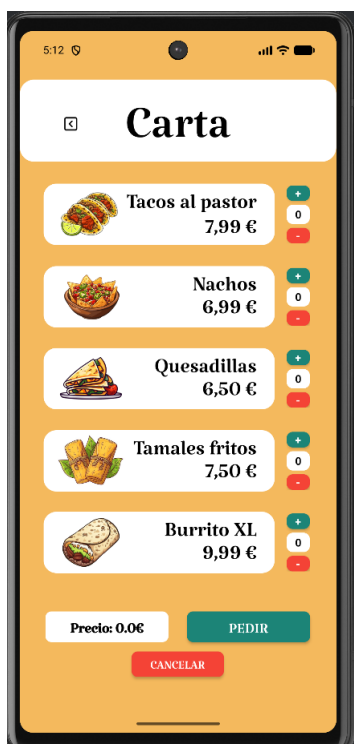
Esta diferenciación se encuentra en la parte baja de la pantalla, la cual se destaca con dos colores diferentes.

En pantalla se encuentran los botones que llevan a los activities de sus respectivas mesas y un botón para volver al activity anterior.

CASO 2: Mesas ocupadas

Mismo funcionamiento que en el caso anterior, pero sin embargo en esta pantalla se pueden apreciar que tanto la **Mesa 3** como la **Mesa 4**, se encuentran ocupadas.

La app no va a permitir su ocupación ni utilización de las mismas, es decir, quedan bloqueadas hasta que desde la APP-Escritorio se libren cobrando el pedido a los clientes.



Carta

Esta es la pantalla en la que se muestran las mesas disponibles del establecimiento. Se trata de un menú con 5 mesas:

- 3 interiores
- 2 terraza

Esta diferenciación se encuentra en la parte baja de la pantalla, la cual se destaca con dos colores diferentes.

En pantalla se encuentran los botones que llevan a los activities de sus respectivas mesas y un botón para volver al activity anterior.

4. APP Escritorio

Inicio app

Esta es la pantalla inicial de la app escritorio, en la que el pulsar en **Entrar**, abre el siguiente activity. Cabe decir que es una app únicamente para el establecimiento de Taco Paco, ya que sus posibles usos son:

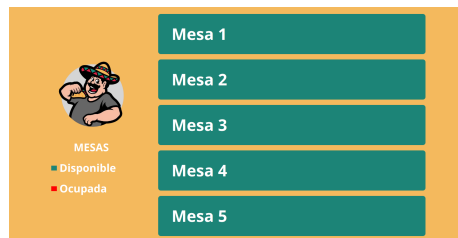
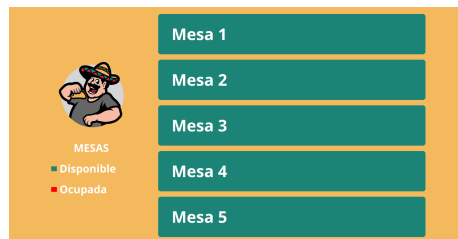
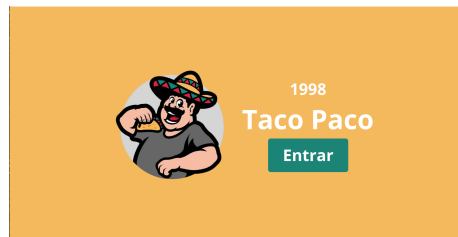
- Establecer un control de qué mesas están o no disponibles
- En caso de que no estén disponibles, se les puede cobrar su pedido desde esta app, lo que libera a continuación la mesa

Control de mesas. CASO 1: Todas libres

Esta pantalla es la principal de la app. Como se puede observar contiene como contenido prioritario las 5 mesas del establecimiento. En este caso no hay ninguna mesa con un pedido pendiente.

Control de mesas. CASO 2: Mesas ocupadas

Las mesas ocupadas se pueden apreciar por su fondo en rojo y el pedido que se debe en su interior. Para liberarla solo hay que clicar en el **botón azul** de cobrar



5. API Rest

5.1. Flujo de la API con las APPs

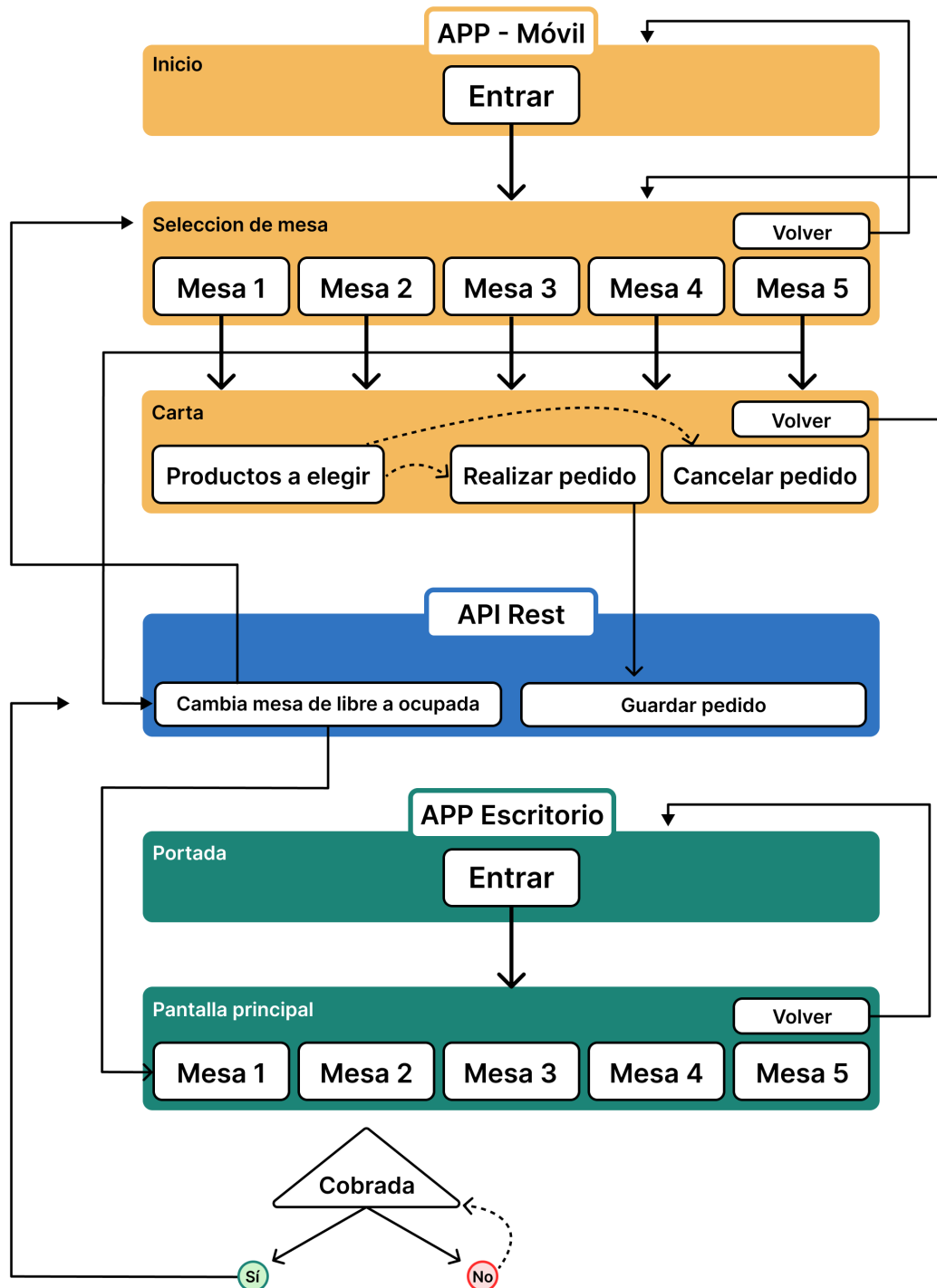


Figura 4: Diagrama de grafo

5.2. Componentes



Figura 5: Estructura de la API

5.2.1. Models/

Explicación de lo que son los models

Modelo Mesa

```
const mongoose = require('mongoose')
const mesaSchema = new mongoose.Schema(
  {
    nombre: { type: String, required: true, unique: true },
    ocupada: { type: Boolean, default: false }
  },
  { versionKey: false }
);
const Mesa = mongoose.model("Mesas", mesaSchema);
```

Modelo Producto

```
const mongoose = require('mongoose')
const pedidoSchema = new mongoose.Schema(
  {
    precioTotal: { type: Number, required: true }
  },
  { versionKey: false }
);
const Pedido = mongoose.model("Pedidos", pedidoSchema);
```

6. Conclusión

7. Bibliografía

- Formatos de imagen de Latex: <https://manualdelatex.com/tutoriales/figuras>
- Imagen al lado de texto en Latex: <https://foro.rinconmatematico.com/index.php?topic=114810.0>
- Texto al rededor de figuras en Lated: https://es.wikibooks.org/wiki/Manual_de_LaTeX/Inclusi%C3%B3n_de_gr%C3%A1ficos/Texto_alrededor_de_figuras
- Insertar código en el pdf: <https://trspos.com/insertar-bloque-de-codigo-latex/>