

# MAMITAS BAGS



# WHAT ARE WE DOING HERE?

NIGHTCLUBS HAVE A LOT OF MOVEMENT, WHETHER IT BE PEOPLE, MONEY, PRODUCTS, AMONG OTHERS. OUR GOAL IS TO BE ABLE TO PROVIDE AN ORDERLY MANAGEMENT AND TO BE ABLE TO PROVIDE A QUALITY SERVICE.

# BENEFITS

## Employee Management

Employee registration (name, gender, age, schedule).  
Shift and availability control.

- ◆ Beverage and Product Management

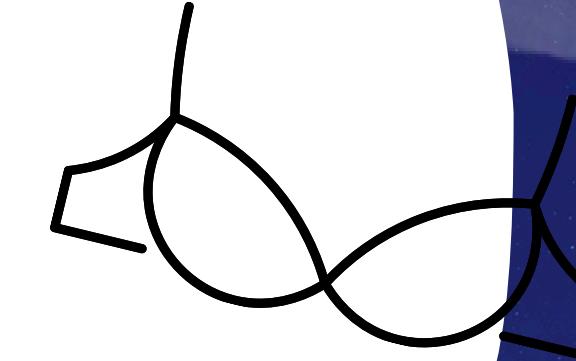
Beverage registration with prices and stock.  
Sales and billing control.

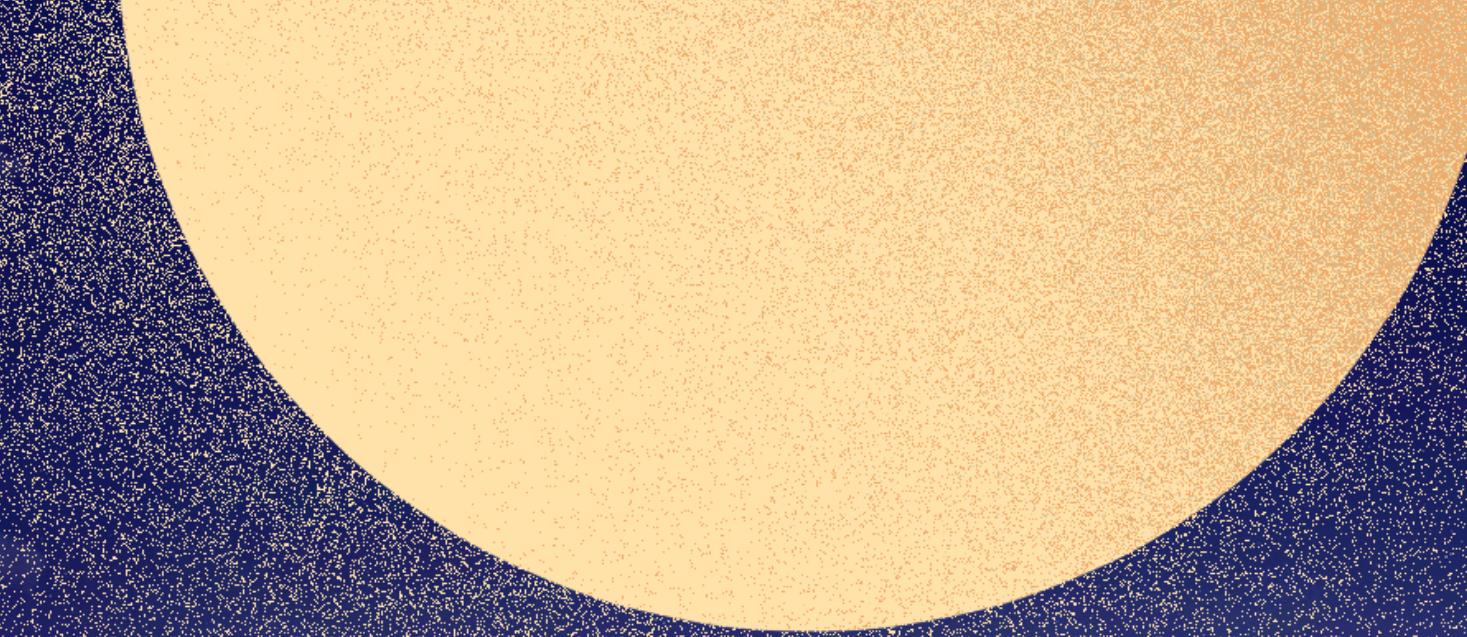
- ◆ Customer Experience

Tracking preferences and consumption.  
Creation of personalized memberships and promotions.

- ◆ Security and Control

Reduction of fraud and theft.  
Restricted access to sensitive information.





We use a Mongo database because of its flexibility and scalability, as well as its easy integration with web and mobile applications.



# DATA BASE

# CODE

```
// controlador.js
const mongoose = require('mongoose');
const { Trabajador, Bebida } = require('../models/personas.js');

mongoose.connect('mongodb://localhost:27017/burdelDB')
  .then(() => console.log('Conectado a MongoDB'))
  .catch((err) => console.error('Error al conectar a MongoDB:', err));

class Controlador {
  async agregarTrabajador(nombre, genero, edad) {
    try {
      const trabajador = new Trabajador({ nombre, genero, edad });
      await trabajador.save();
      console.log('Trabajador guardado:', trabajador);
    } catch (error) {
      console.error('Error al guardar trabajador:', error);
    }
  }

  async agregarBebida(nombre, precio) {
    try {
      const bebida = new Bebida({ nombre, precio });
      await bebida.save();
      console.log('Bebida guardada:', bebida);
    } catch (error) {
      console.error('Error al guardar bebida:', error);
    }
  }
}

const mongoose = require('mongoose');
// Esquema de Trabajador
const trabajadorSchema = new mongoose.Schema({
  nombre: { type: String, required: true },
  genero: { type: String, enum: ['F', 'M'], required: true },
  edad: { type: Number, required: true },
});

const Trabajador = mongoose.model('Trabajador', trabajadorSchema);

const bebidaSchema = new mongoose.Schema({
  nombre: { type: String, required: true },
  precio: { type: Number, required: true },
});

const Bebida = mongoose.model('Bebida', bebidaSchema);

module.exports = { Trabajador, Bebida };
```

Code Blame 7 lines (6 loc) · 240 Bytes

```
1 import mongoose from 'mongoose';
2 const bebidaSchema = new mongoose.Schema({
3   nombre: { type: String, required: true },
4   precio: { type: Number, required: true },
5 });
6
7 export const Bebida = mongoose.model('Bebida', bebidaSchema);
```

```
1 import mongoose from "mongoose";
2
3 // Esquema de Trabajador
4 const trabajadorSchema = new mongoose.Schema({
5   nombre: { type: String, required: true },
6   genero: { type: String, enum: ['F', 'M'], required: true },
7   edad: { type: Number, required: true },
8 });
9
10 export const Trabajador = mongoose.model('Trabajador', trabajadorSchema);
```



!Mamitas Ags;

# CODE

```
_id: ObjectId('67d30f1c1eb0fff32683db1d')
nombre : "dick"
precio : 222
__v : 0
```

```
_id: ObjectId('67d30e011eb0fff32683db15')
nombre : "dd"
genero : "F"
edad : 58
__v : 0
```

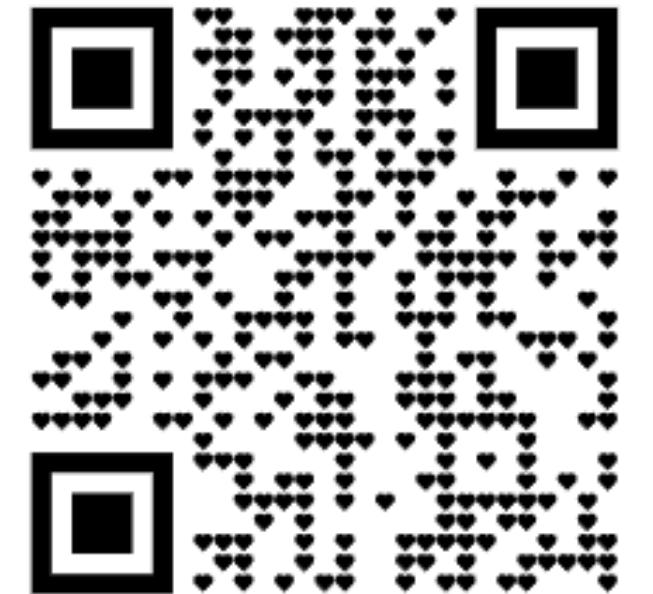
```
_id: ObjectId('67d30e031eb0fff32683db17')
nombre : "dd"
genero : "F"
edad : 58
__v : 0
```

```
_id: ObjectId('67d30ea01eb0fff32683db19')
nombre : "dd"
genero : "M"
edad : 58
__v : 0
```

▼ ⚡ ClubNocturno

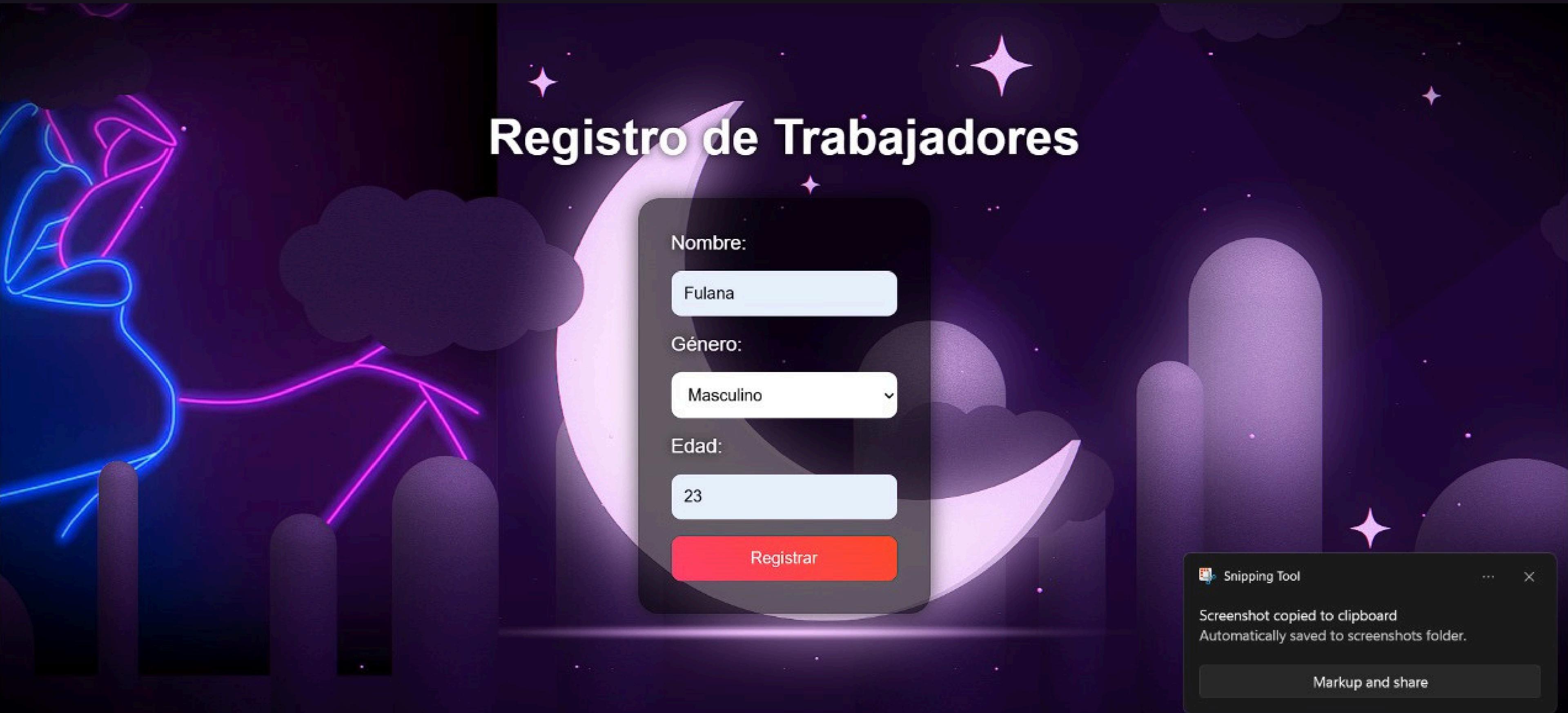
📁 bebidas

📁 trabajadores



!Mamitas Ags!

For improved control and quality of service, we also set up a website..





THANKS

GIRLS