

Proyecto de fin de ciclo

Documentación

2º CFGS Desenvolvemento de Aplicacións Multiplataforma

Realizado por:

Daniel Pérez Ferreiro

Índice

Introducción	3
Tecnologías usadas.....	3
Arquitectura.....	5
Análisis de requisitos	6
Diseño estático	7
Modelo de datos	7
Viabilidad.....	8
Calidad	9
Conclusiones y mejoras futuras.....	9
Bibliografía	9

Introducción

En este documento se explicarán diferentes aspectos relacionados con el proyecto de fin de ciclo (PFC) realizado, primero comenzaremos contextualizando sobre el proyecto. El proyecto se basa en la automatización de procesos en ambientes gastronómicos, en específico restaurantes, aunque puede modificarse para que funcione en más lugares, específicamente este programa tiene como intención automatizar la comunicación entre los camareros, la cocina y la barra; esta comunicación se conseguiría mediante el uso de programas conectados a un api que estaría funcionando en el servidor central, ayudando en la comunicación entre dispositivos como móviles y ordenadores.

El desarrollo de la aplicación se realiza como autónomo ya que me da la ventaja a diferencia de ser una empresa de poder tener una estructura administrativa mas sencilla (lo hago yo todo), aun con las consecuencias que conlleva esto, tales como mayor tasa de impuestos a pagar, no tener una estabilidad monetaria, etc.

Tecnologías usadas

SQL

- ¿Que es SQL?

SQL es un acrónimo para las siglas (Structure Query Language), y es un lenguaje que nos permite gestionar de diferentes maneras las bases de datos, las bases de datos SQL almacena su información en tablas, las cuales cuentan con filas y columnas que representan los datos que hay dentro de la tabla y los tipos de datos que son.

- ¿Por qué utilizar SQL?

El porqué de la utilización de SQL frente a NoSQL es debido a que tengo mas soltura en el lenguaje SQL, y que ya se nos fue explicado este lenguaje con más profundidad en el ciclo y esto permite reducir los tiempos de desarrollo.

- ¿En qué se utiliza SQL?

Dentro de este proyecto el lenguaje SQL es utilizado para la creación y mantenimiento de la base de datos utilizando el SGBD de SQL Server el cual también se nos fue explicado en el ciclo con lo cual brinda de nuevo la ventaja de reducir los tiempos de desarrollo

Java

- ¿Que es Java?

Java es un lenguaje de programación desarrollado por Sun Microsystems en 1995, actualmente fue adquirido por Oracle. Es un lenguaje de maquina virtual, lo cual quiere decir que se puede ejecutar en cualquier dispositivo que tenga la maquina virtual de Java, de esta manera el desarrollo es mas sencillo porque puedes utilizar el mismo código para dos sistemas operativos diferentes.

- ¿Por qué utilizar Java?

La utilización de Java en este proyecto es debido a dos factores principales, el primero es la facilidad de desarrollo en este lenguaje de programación debido a las practicas hechas en clase, y el segundo factor es que para realizar la api se ha utilizado Springboot el cual entre los lenguajes que nos ofrece para el desarrollo de la api se encuentra Java el cual como en el punto anterior dije me desenvuelvo mejor.

Git

- ¿Que es Git?

Git es un sistema de control de versiones creado por Linus Torvalds, el cual se diferencia de los demás sistemas de control de versiones de la época principalmente en que este nos permite guardar los cambios entre versiones y no todo el contenido, permitiéndonos optimizar el espacio entre versión y versión, otra diferencia era que los anteriores programas no permitían que pusieses a otras personas en el mismo proyecto, lo cual lo soluciono también Git. En este proyecto se utiliza específicamente GitHub Desktop el cual es una herramienta que nos permite tener una interacción mas sencilla con el sistema de guardado en la nube de GitHub.

- ¿Por qué utilizar Git?

El uso de Git en este proyecto es para poder mantener un control de versiones organizado de las aplicaciones y poder tener el proyecto en la nube mediante GitHub

Draw.io

-¿Qué es Draw.io?

Draw.io es una herramienta de Google que nos permite realizar diagramas online y guardarlos tanto en Drive como en local, nos permite realizar diagramas de flujo, uml, red, etc

-¿Por qué utilizar Draw.io?

Gracias a esta herramienta se puede realizar de manera mas sencilla un diagrama E/R para la base de datos

Arquitectura

Para este proyecto se han utilizado las siguientes arquitecturas:

·Arquitectura cliente-servidor:

-Descripción:

La arquitectura-cliente servidor es un modelo por el cual los equipos tienen la opción de ser rol cliente o servidor. Los clientes son equipos los cuales utilizan servicios que ofrecen los servidores, estos servicios pueden ser variados, tales como pasar imágenes, entregar direcciones ip, etc.

-Uso en el proyecto:

Esta arquitectura se ha utilizado en dos lugares, el primero es en el api, que, aunque se ejecute en la misma máquina que la base de datos, se le especifica por código donde se encuentra la base de datos a la cual le va a pedir ciertos datos de tablas; y el otro lugar es en la aplicación, ya que en ella se especifica la dirección que ha de apuntar para acceder a la api.

-Ventajas:

- Centralización de los recursos
- Escalabilidad
- Sencillez de mantenimiento

·Modelo vista controlador:

-Descripción:

El modelo vista controlador es una arquitectura de desarrollo de software que permite separar el programa en tres tipos de componentes (Vista, Models, Controladores), y de esta manera podemos separar la lógica de la aplicación en diferentes partes.

-Uso en el proyecto:

Este tipo de arquitectura se ha utilizado en el proyecto para el desarrollo de la api, de esta manera en caso de querer añadir mas objetos o modificar el funcionamiento de la aplicación no habría que cambiarlo todo.

-Ventajas:

- Separación de los tipos de lógica.
- Sencillez para crear diferentes representaciones de los mismos datos
- Facilidad para realizar pruebas unitarias
- Reutilización de componentes

Análisis de requisitos

Captación de requisitos:

Información relativa a la realización del proyecto:

El desarrollo de este proyecto se realizaría utilizando una metodología ágil la cual nos permite tener una comunicación mas cercana con el cliente, teniendo pequeñas reuniones cada poco para poder acordar si como esta yendo el proyecto es igual a como lo pensaba el cliente.

Las fases de este proyecto se verán reflejadas a continuación en el siguiente calendario:

2 de marzo a 2 de abril	3 de mayo a 2 de junio	5 de junio	6 de junio en adelante
Captacion de requisitos con el cliente	Desarrollo de la aplicacion	Exposicion al cliente	Soporte

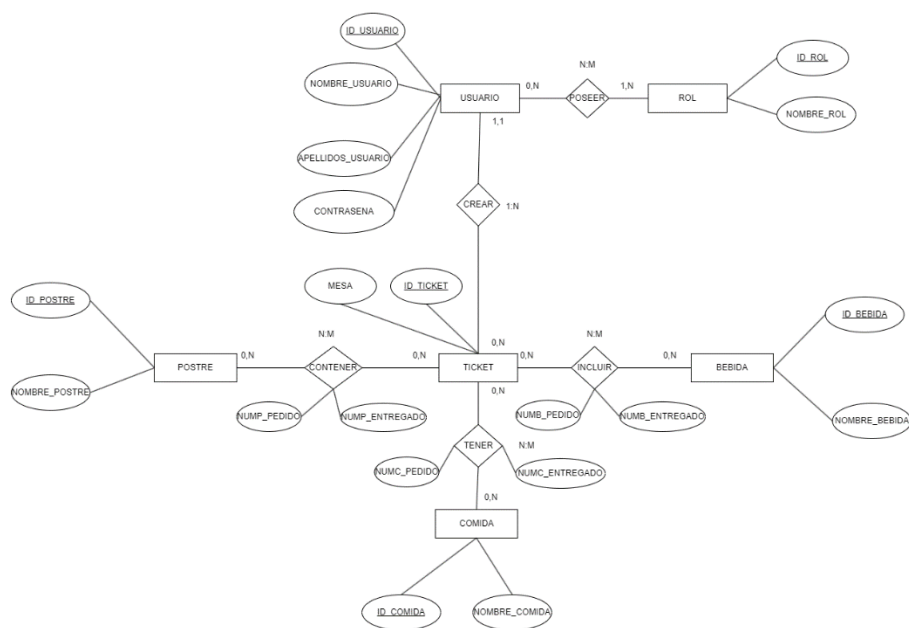
Se entiende que a continuación de la última fecha se seguirá dando soporte mensualmente siempre que se cumplan con los plazos de cobro.

Diseño estático

Modelo de datos

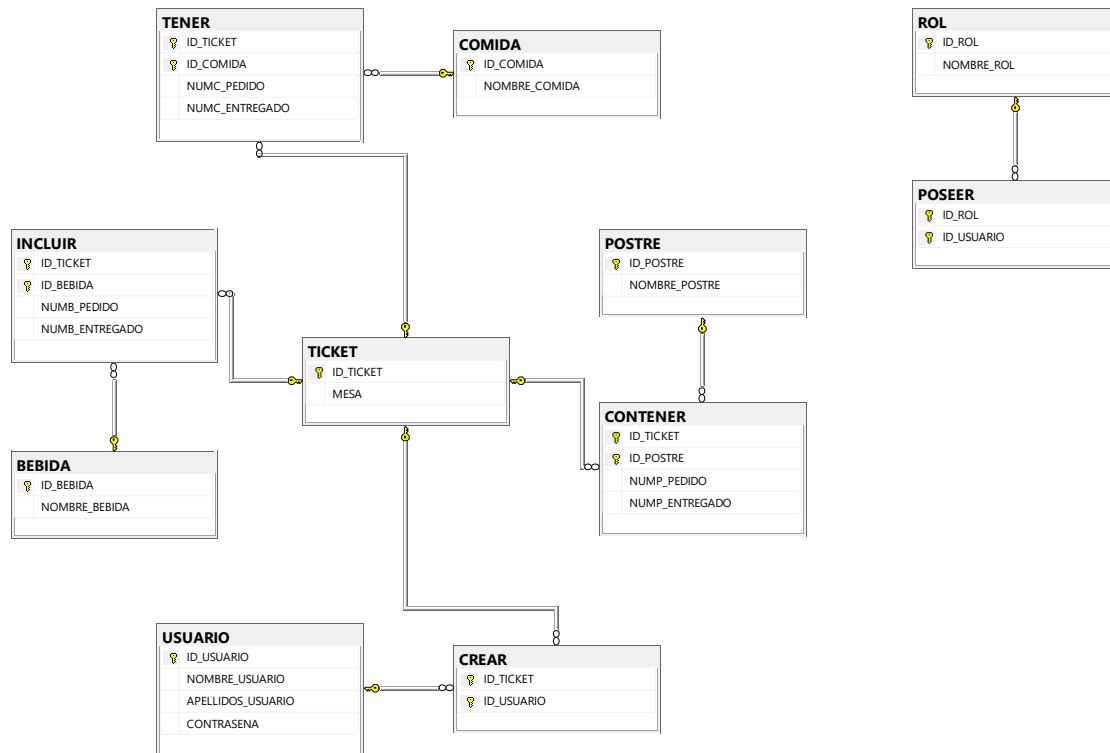
A continuación, se mostrarán el modelo E-R mediante el cual se ha realizado la base de datos:

-Modelo E-R:



A continuación se mostrara el diagrama de tablas sacado después de la creación de la base de datos:

-Diagrama de tablas:



Viabilidad

Estudio de mercado:

Después de investigar el mercado de las aplicaciones de TPV en sectores como la hostelería se han podido ver dos bandos muy claros, el primero es las aplicaciones profesionales con soporte cuyos precios rondan sobre los 200€ al mes los cuales ofrecen la instalación y el soporte de errores; y luego están los programas gratuitos los cuales no ofrecen soporte ni instalación. Como ejemplo de los programas podemos ver: Ágora TPV, SumUp...

Análisis económico:

Como este proyecto esta desarrollado de manera individual, y está planteado como autónomo, a los ingresos adquiridos habrá que quitarle el 24% ya que la idea es tener una media de ganancias de 15.000€ al año, para lograr esto mi idea es cobrarle 120€ de mensualidad a las empresas por el soporte e instalación; adicionalmente se cobrara un plus en caso de querer añadir cosas al programa que no están. Obteniendo un beneficio neto de 11.400€.

Estudio de riesgos:

Uno de los principales riesgos de este proyecto es que en caso de que la popularidad y uso de este producto flaquee un poquito ya las ganancias se verán afectadas, adicionalmente el mercado en la zona de Galicia ya esta muy dominado por las mismas 4 empresas, con lo cual la posibilidad del fracaso es probable.

Calidad

Conclusiones y mejoras futuras

Como principal conclusión de este proyecto indico que la idea principal era tener muchas mas cosas, tales como la aplicación móvil, pero en el tiempo establecido y con las practicas no dio tiempo a más.

Mejoras a futuro:

- Creación de aplicación móvil
- Modificación base de datos (precios, ingredientes...)
- Control de versiones
- Pago

Bibliografía

[\(\[Pagina sobre que es la arquitectura cliente servidor\], s.f.\)](#)

[\(\[Explicación modelo vista controlador\], s.f.\)](#)

[\(\[Ventajas del MVC\], s.f.\)](#)

[\(\[Ventajas del modelo cliente-servidor\], s.f.\)](#)

[\(\[Explicación git\], s.f.\)](#)

[\(\[Explicación Java\], s.f.\)](#)

[\(\[Que es SQL\], s.f.\)](#)

