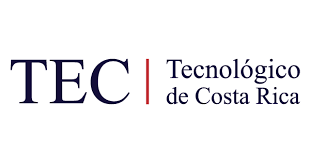
**Instituto Tecnológico de Costa Rica**

**Área Académica de Ingeniería en Computadores**

**Programa de Licenciatura en Ingeniería en Computadores**

**Curso: CE4101 Especificación y Diseño de Software**



**EatOnTimeTEC**

**Documento de diseño**

**Realizado por:**

Bryan Badilla Amador 2016116481

Emanuel Esquivel Lopez 2016133597

Vinicio Monge Avendaño 2016128536

Álvaro Quesada Ramírez 2016005829

**Profesor:**

**Luis Diego Noguera Mena**

**Fecha: Noviembre 01, 2019**

****

**Documento de Diseño**

**Empresa: X-TEC**

**Proyecto: EatOnTimeTEC**

**Contactos con X-TEC: Luis Diego Noguera Mena**

**Equipo de trabajo:**

**Bryan Badilla Amador**

**Emanuel Esquivel López**

**Vinicio Monge Avendaño**

**Álvaro Quesada Ramírez**

**Noviembre 01, 2019**

# Índice

[**Índice**](#_6jaxkti4vfco) **2**

[**Introducción**](#_mkc7jpjc7z58) **3**

[1.1 Control de documento](#_3zsqkstjzlyu) 3

[1.2 Tabla de definiciones y Acrónimos](#_ftohv0ju6r9f) 4

[1.3 Visión general del documento](#_i9lo62o7707e) 6

[**Descripción de la Arquitectura**](#_w8vwqpak6js2) **7**

[2.1 Diagrama de Contexto con tabla de descripción](#_wtguikcypzdo) 7

[2.2 Modelo de estructura modular](#_aq5sbwladoc6) 8

[**Requerimientos Funcionales**](#_5pke4sehvfys) **10**

[3.1 Diagrama de funcionalidades](#_m8iifaolw0vi) 10

[3.2 Diagramas de Secuencia de clases](#_n3nsssn58iiv) 11

[3.3 Diagrama detallado de Clases de diseño](#_npjuw7fpm16c) 16

[3.4 Diagrama Completo de paquetes](#_hrol4k2f7rqn) 17

[3.5 Explicación de patrones de diseño utilizados](#_239anfloqdpr) 18

[**Diseño de Datos**](#_ntf5tkx0wngu) **19**

[4.1 Diagrama de Base de Datos](#_ri5de6s83d39) 19

# 

# Introducción

## 1.1 Control de documento

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Responsable | Descripción de cambios realizados | Fecha |
| 1.0 | Emanuel, Bryan, Vinicio, Álvaro | Se finaliza el documento de diseño. | 01/11/19 |
| 2.0 | Emanuel, Bryan, Vinicio, Álvaro | Se corrigen los diagramas y se intercambia un patrón de diseño. | 21/11/19 |

**Tabla 1.Control del documento**

## 

## 1.2 Tabla de definiciones y Acrónimos

|  |  |
| --- | --- |
| **Concepto** | **Definición** |
| EatOnTimeTEC. | Nombre del proyecto y aplicación a desarrollar. |
| Usuario. | Todas las personas que interactúan con el sistema. Lo constituyen los usuarios registrados y no registrados. |
| Cliente. | Es el usuario principal de la aplicación. Utiliza el sistema para realizar órdenes, acumular puntos por orden, cambiar puntos por orden y visualizar el dashboard con información personal. |
| PR. | Código de user stories que representan requerimientos relacionados al registro de usuarios. |
| PG. | Código de user stories que representan requerimientos relacionados a la gestión del perfil del usuario. |
| A. | Código de user stories que representan requerimientos relacionados a las amistades de los usuarios en el sistema. |
| VM. | Código de user stories que representan requerimientos relacionados a la visualización del menú. |
| RO. | Código de user stories que representan requerimientos relacionados a realizar ordenes. |
| GO. | Código de user stories que representan requerimientos relacionados a gestionar órdenes. |
| CA. | Código de user stories que representan requerimientos relacionados a realizar calificaciones. |
| GM. | Código de user stories que representan requerimientos relacionados a gestionar menú. |
| P. | Código de user stories que representan requerimientos relacionados a los puntos por orden. |
| D. | Código de user stories que representan requerimientos relacionados a los dashboards. |
| RA. | Código de user stories que representan requerimientos relacionados al reporte de auditoría. |
| RC. | Código de user stories que representan requerimientos relacionados a recibir recomendaciones. |
| F. | Código de user stories que representan requerimientos relacionados a morosidades. |

**Tabla 2. Definiciones y acrónimos.**

## 

## 1.3 Visión general del documento

El presente documento tiene como principal objetivo establecer una guía para la codificación e implementación de la aplicación EatOnTimeTEC propuesta por la empresa X-TEC.

Con este documento se pretende establecer todos los detalles del diseño, de la arquitectura y los componentes que conforman la aplicación para posteriormente proceder con el desarrollo.

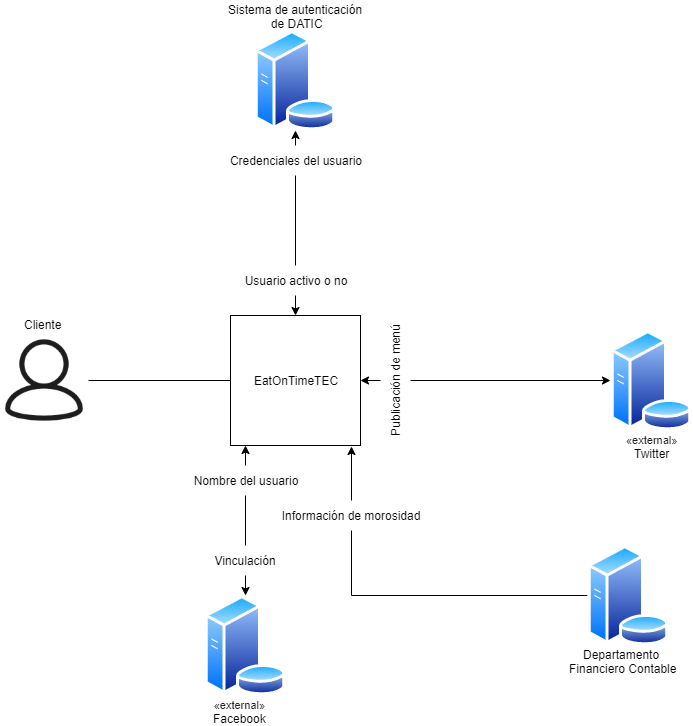
Este documento tiene como objetivo servir como acuerdo entre el grupo de diseño, el grupo de programación, el grupo de pruebas y X-TEC, como la aplicación va a ser implementada.

En el documento se encuentran las siguientes secciones:

* Control de documento.
* Tabla de definiciones y acrónimos.
* Diagrama de contexto.
* Modelo de estructura modular.
* Diagrama de funcionalidades.
* Diagrama de secuencia de clases detallado para los escenarios de cambio de puntos, gestión de amigos y órdenes colaborativos.
* Diagrama detallado de clases de diseño.
* Diagrama de paquetes.
* Explicación de los patrones de diseño implementados en el proyecto.
* Diagrama de bases de datos.

# Descripción de la Arquitectura

## 2.1 Diagrama de Contexto con tabla de descripción

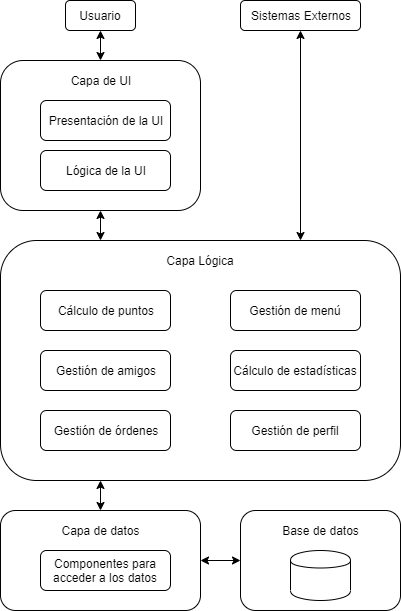


**Figura 1. Diagrama de contexto de la aplicación EatOnTimeTEC.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requerimientos | Cliente | Puntos | Financiero | DATIC | Facebook |
| Cliente |  | Utiliza | Endeuda | Auténtica | Vincula |
| Puntos | Son usados |  |  |  |  |
| Financiero | Adeuda |  |  |  |  |
| DATIC | Auténtica |  |  |  |  |
| Facebook | Vincula |  |  |  |  |

**Tabla 3. Descripción del diagrama de contexto.**

## 2.2 Modelo de estructura modular



**Figura 2. Modelo de estructura modular de la aplicación EatOnTimeTEC.**

**Capas utilizadas:**

**Capa de UI**: Esta es la capa con la que el usuario va interactuar directamente. Contiene la lógica de la interfaz gráfica y se visualiza de forma entendible para el usuario.

**Capa lógica**: Esta capa contiene toda la lógica que debe ejecutar la aplicacion y que es necesaria para la realización de funciones de parte de los usuarios. Es la capa más “gruesa” del sistema debido a que interactúa prácticamente con todos los componentes del mismo sistema.

**Capa de datos**: Esta capa contiene toda la lógica para poder interactuar con los datos que se deben acceder o guardar en la base de datos del sistema.

# Requerimientos Funcionales

## 3.1 Diagrama de funcionalidades

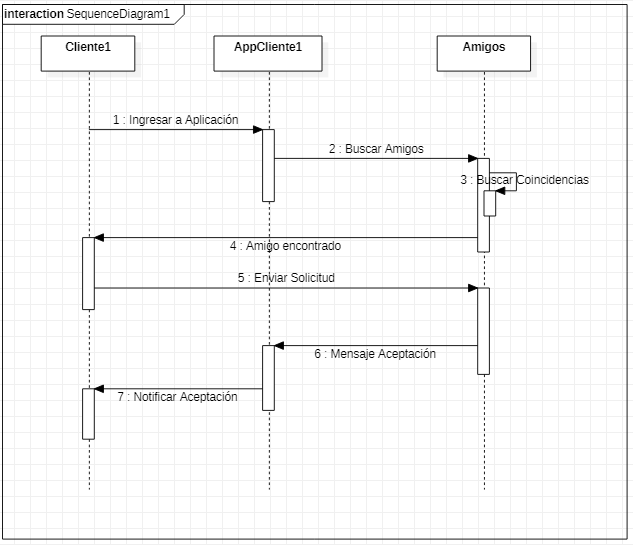


**Figura 3. Diagrama de funcionalidades.**

## 3.2 Diagramas de Secuencia de clases

**Diagrama de secuencia agregar amigos**

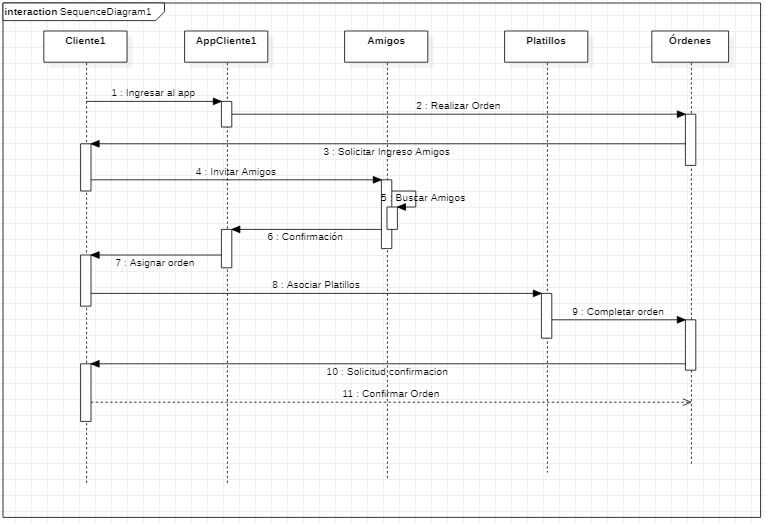
En el siguiente diagrama se puede ver si el cliente quiere agregar como amigo a otro usuario del sistema, se puede ver que que se toma en cuenta los datos de búsqueda (nombre o teléfono) y se muestra la aprobación de la misma.



**Figura 4. Diagrama de secuencia Agregar amigos.**

**Diagrama de secuencia realizar orden en conjunto**

En el siguiente diagrama se puede ver como el cliente realiza una orden el conjunto con un amigo, se puede ver que el amigo acepta la misma, y el usuario o cliente que realiza la orden puede agregar platillos a la orden, asignarlos y completar la orden.

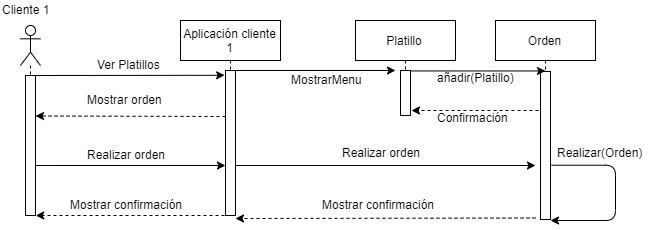


**Figura 5. Diagrama de secuencia Realizar orden en conjunto.**

Gestionar amigos, deben describir relaciones entre objetos. En el de generar orden falta clase menú y el ciclo para agregar más platillos, falta objeto Usuario para sumarle los puntos. En el de cambio de puntos faltan objeto de usuario para saber los puntos y objeto Tasa de conversión para poder hacer el intercambio.

**Diagrama de secuencia realizar orden en solitario**

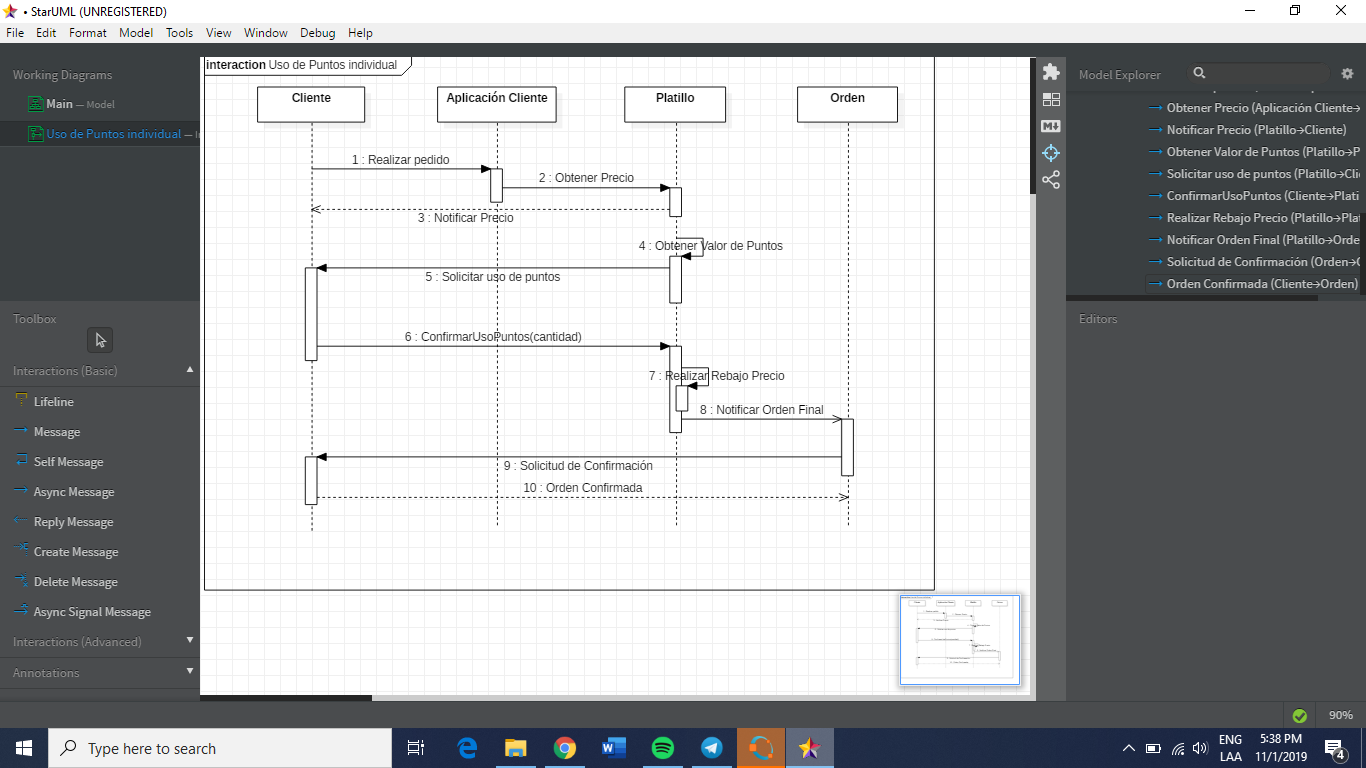
En el siguiente diagrama se puede ver como el cliente realiza una orden el solitario, el cliente que realiza la orden puede agregar platillos a la orden y completar la orden.



**Figura 6. Diagrama de secuencia realizar orden en solitario.**

**Diagrama de secuencia uso de puntos en solitario**

En el siguiente diagrama se muestra el proceso correspondiente al uso de puntos con la finalidad de conseguir un rebajo parcial o total del precio del platillo correspondiente a su orden.

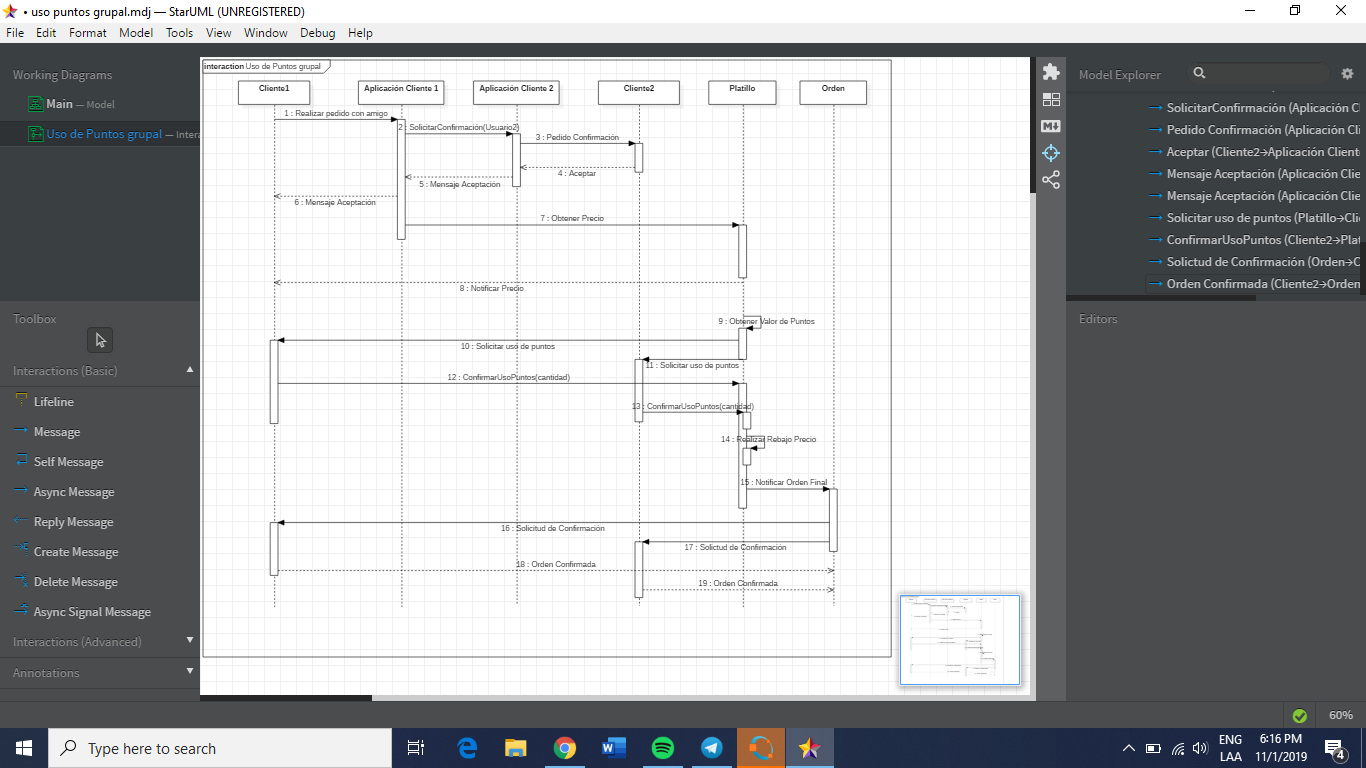


**Figura 7. Diagrama de secuencia uso de puntos en solitario.**

## 

**Diagrama de secuencia uso de puntos con amigos**

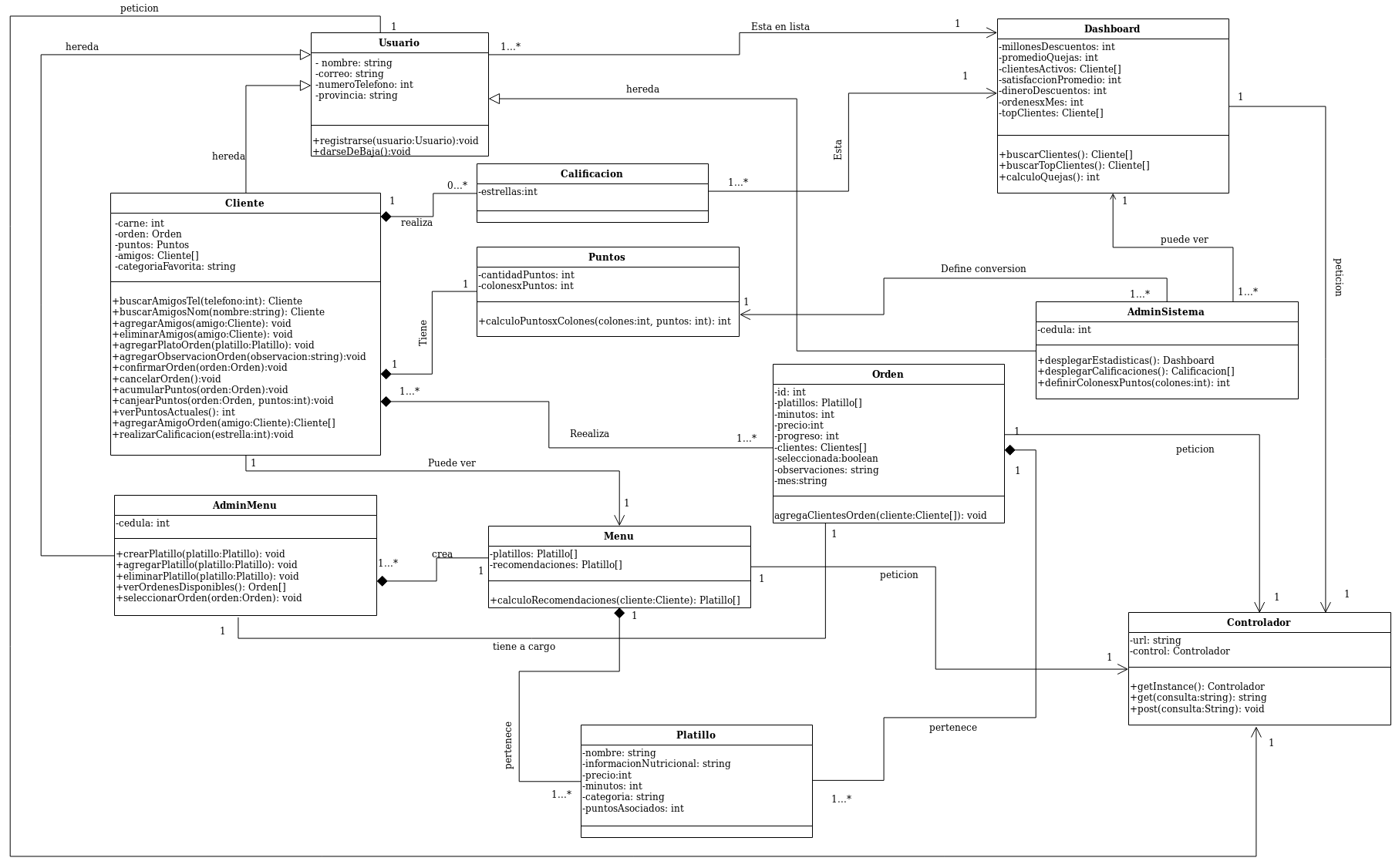
En el siguiente diagrama se muestra el proceso correspondiente al uso de puntos en conjunto con otras personas invitadas a la orden. Esto se realiza con la finalidad de conseguir un rebajo parcial o total del precio del platillo correspondiente a su orden por medio de un canjeo de puntos. Se puede observar como varios usuarios accede al uso de los mismos.



**Figura 8. Diagrama de secuencia uso de puntos con amigos.**

## 

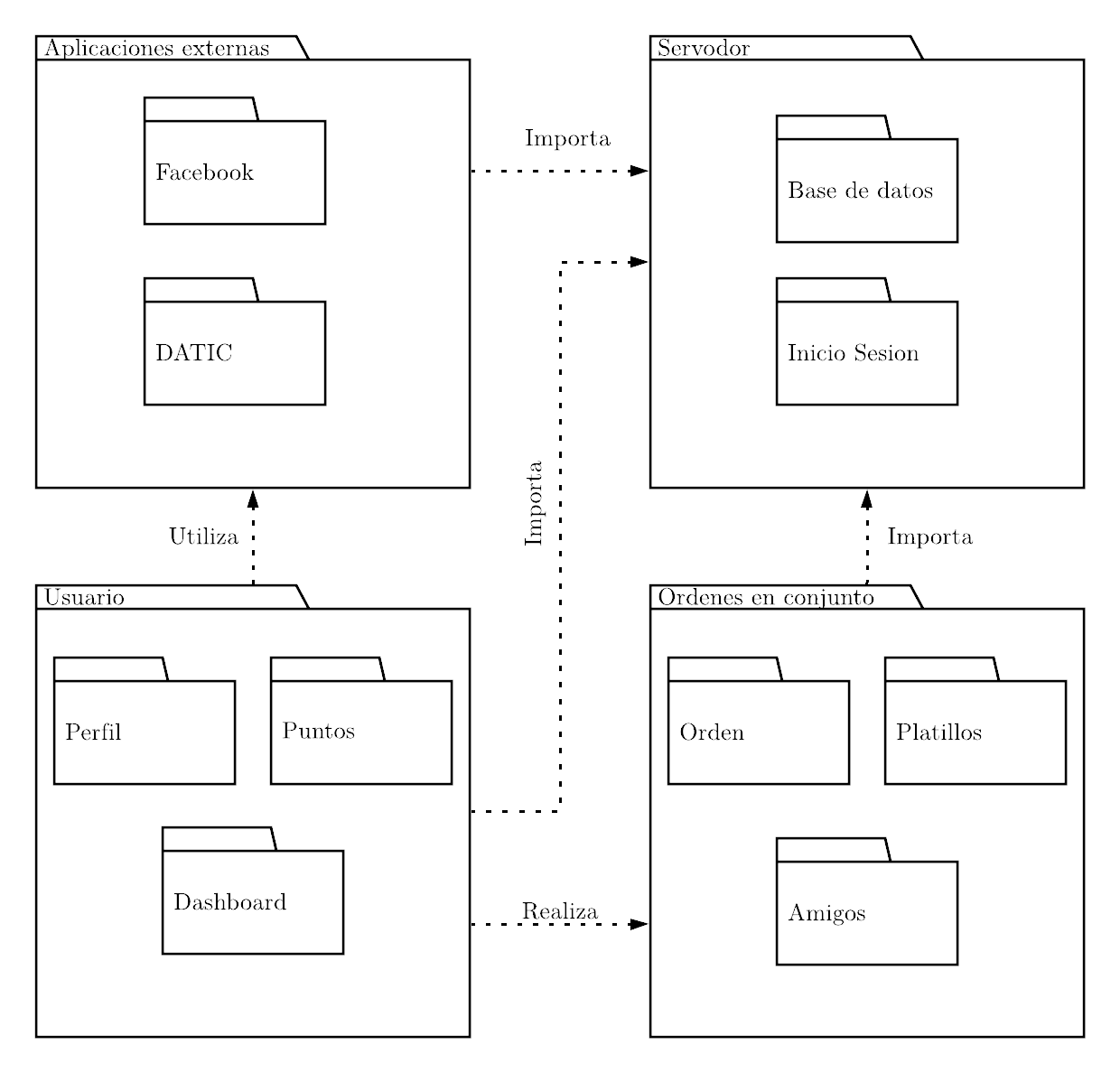
## 3.3 Diagrama detallado de Clases de diseño



**Figura 9. Diagrama de clases del sistema.**

## 

## 3.4 Diagrama Completo de paquetes



**Figura 10. Diagrama de paquetes.**

## 

## 3.5 Explicación de patrones de diseño utilizados

**Singleton**: Este patrón permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención es que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

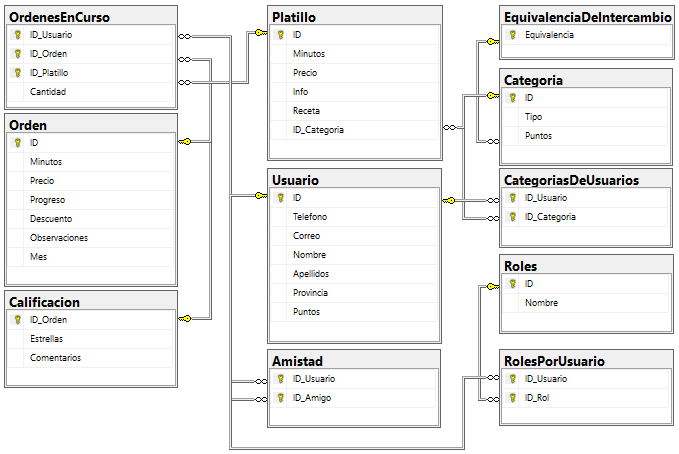
En el diseño del sistema EatOnTimeTEC se utiliza en la clase que realiza las peticiones al servidor, la cual es llamada **Controlador**. Esta clase es utilizada por la mayoría de las clases del sistema, es mucho mejor que exista un único objeto compartido por todas las clases que lo utilizan.

**Prototype**: este patrón es usado para permitir la creación de una variedad de objetos idénticos, se necesitan crear distintos machotes para la inserción y manipulación de información de la base de datos en e el servidor.

En el diseño del sistema EatOnTimeTEC se utiliza en los esquemas al crear un objeto nuevo y asignarle los datos que se desean incorpora, rescatar u operar.

# Diseño de Datos

## 4.1 Diagrama de Base de Datos



**Figura 11. Diagrama Base de Datos.**