

DISEÑO Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

Sistema de gestión para el restaurante Pomodoro

Alumnos:

Nikolov Vasilev, Vladislav
Román Guerrero, Nazaret
Sánchez Guerrero, José María

Grupo: 3ºA1

Profesor: Daniel Sánchez Fernández

Descripción del sistema

Nuestro sistema de información va a ser sobre la gestión y mantenimiento de un restaurante familiar de comida italiana llamado "*Pomodoro*". El sistema se va a dividir en los tres subsistemas siguientes:

- Subsistema de gestión de empleados.
- Subsistema de gestión de menús.
- Subsistema de gestión de proveedores.

El subsistema de gestión de empleados se va a encargar de las siguientes funciones:

- Incluir un nuevo empleado en el sistema, mediante su nombre y apellidos, fecha de nacimiento, DNI, teléfono, dirección, estado civil, correo electrónico, formación, número de cuenta bancaria y número de seguridad social. A partir de estos datos, el sistema asignará como identificador del empleado el DNI, que se almacenará junto con los datos proporcionados por el usuario.
- Eliminar un empleado, mediante su DNI. Cuando se elimina un empleado se borrarán todos los datos asociados a éste.
- Modificar un empleado, utilizando el DNI, el dato o los datos a modificar y la nueva información. No se cambiará ningún otro dato que no sean los especificados.
- Buscar un empleado, mediante su DNI. Se devolverá la información completa del empleado buscado.
- Asignar tareas, poniendo la tarea a asignar y el DNI del empleado que la realiza. Esta información se almacenará en una lista que contiene las tareas que realizan los empleados. El identificador será la tarea, ya que cada tarea puede tener varios empleados (por ejemplo, servir mesas: 12345678W, 87654321Q).
- Modificar tareas, utilizando el DNI del empleado y la nueva tarea a la que lo quieres asignar. Si se asigna un empleado a una tarea que ya está realizando, lo eliminará de la lista y dejará de hacer esa tarea.

El subsistema de gestión de menús se va a encargar de las siguientes funciones:

- Incluir un nuevo plato en el sistema, mediante su nombre, precio, ingredientes (cadena alfanumérica de hasta 50 caracteres), productos alérgenos (cadena de hasta 50 caracteres) y su disponibilidad (cadena de 2 caracteres). Para cada plato se generará un código de plato (cadena numérica de 4 dígitos) que permitirá identificarlo.

- Eliminar un plato, mediante su código de plato. Se elimina definitivamente del sistema toda la información asociada al plato. Si ese plato forma parte de un menú, se elimina el plato del menú.
- Modificar un plato, mediante su código de plato, la información que se desea modificar y los nuevos datos. Se permite modificar el precio, los ingredientes, los productos alérgenos y la disponibilidad. Si un plato pasa a estar no disponible, se elimina del menú asociado (si tiene).
- Buscar un plato, mediante su código de plato. Permite obtener toda la información relacionada al plato.
- Añadir un nuevo menú, mediante el identificador de menú (cadena numérica de hasta 4 dígitos), los códigos de platos que lo componen, su precio y su disponibilidad (cadena de 2 caracteres).
- Modificar un menú existente mediante su identificador de menú, los datos que se deseen modificar y los nuevos datos. Se permite modificar los códigos de los platos que lo componen (cambiarlos/eliminarlos), el precio y la disponibilidad.
- Eliminar un menú mediante su identificador de menú. Elimina definitivamente la información de ese menú. No elimina la información asociada a los platos incluidos en el menú.

El subsistema de gestión de proveedores se va a encargar de las siguientes funciones:

- Incluir un nuevo proveedor en el sistema, mediante su nombre y apellidos, DNI, teléfono, correo electrónico, nombre de la empresa a la que pertenece, NIF de la empresa y tipo de productos que provee (que se representa con una cadena de 20 caracteres). Con estos datos, el sistema asigna un identificador único a cada proveedor (su DNI) que se dé de alta en el sistema.
- Eliminar un proveedor mediante su DNI. Todos los datos del proveedor se eliminan del sistema.
- Modificar un proveedor, utilizando el dato o los datos a modificar y los nuevos datos. Mediante el identificador único que se le asigna, se busca al proveedor en el sistema y se modifican los datos que se han suministrado. El resto de datos quedan igual.
- Buscar un proveedor mediante el código del ingrediente que provee. Devuelve la información del proveedor.
- Establecer ingredientes provistos mediante el código de ingrediente (una serie de 5 números), el tipo (una cadena de máximo 20 caracteres), el nombre de dicho ingrediente (con una cadena de 25 caracteres) y la cantidad (con un número entero entre 1 y 500). El ingrediente tiene como identificador único el propio código que se introduce.

Especificación de requisitos

El subsistema de gestión de empleados va a tener los siguientes requisitos de datos, funcionales y restricciones semánticas:

→ **RD1.1** Los datos de entrada de un empleado:

- Nombre y apellidos (una cadena de hasta 60 caracteres no vacía).
- Fecha de nacimiento (una cadena de 10 caracteres alfanuméricos del tipo DD/MM/AAAA).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Dirección (una cadena de hasta 120 caracteres no vacía).
- Estado civil (se podrá elegir entre soltero, casado, viudo o divorciado)
- Correo electrónico (una cadena de hasta 60 caracteres).
- Formación (una cadena de hasta 200 caracteres).
- Número de cuenta bancaria (una cadena de hasta 20 caracteres).
- Número de la seguridad social (una cadena de hasta 12 números).

→ **RD1.2** Los datos almacenados de un empleado:

- Nombre y apellidos (una cadena de hasta 60 caracteres no vacía).
- Fecha de nacimiento (una cadena de 10 caracteres alfanuméricos del tipo DD/MM/AAAA).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Dirección (una cadena de hasta 120 caracteres no vacía).
- Estado civil (se podrá elegir entre soltero, casado, viudo o divorciado)
- Correo electrónico (una cadena de hasta 60 caracteres).
- Formación (una cadena de hasta 200 caracteres).
- Número de cuenta bancaria (una cadena de hasta 20 caracteres).
- Número de la seguridad social (una cadena de hasta 12 números).

→ **RD1.3** Los datos de salida de un empleado:

- Nombre y apellidos (una cadena de hasta 60 caracteres no vacía).
- Fecha de nacimiento (una cadena de 10 caracteres alfanuméricos del tipo DD/MM/AAAA).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).

- Dirección (una cadena de hasta 120 caracteres no vacía).
- Estado civil (se podrá elegir entre soltero, casado, viudo o divorciado)
- Correo electrónico (una cadena de hasta 60 caracteres).
- Formación (una cadena de hasta 200 caracteres).
- Número de cuenta bancaria (una cadena de hasta 20 caracteres).
- Número de la seguridad social (una cadena de hasta 12 números).

→ **RD1.4** La lista de tareas almacenada que realizan los empleados:

- Nombre de la tarea (una cadena de hasta 50 caracteres no vacía).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).

→ **RD1.5** Los datos necesarios para eliminar un empleado.

- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).

→ **RD1.6** Los datos necesarios para modificar un empleado.

- Nombre y apellidos (una cadena de hasta 60 caracteres no vacía).
- Fecha de nacimiento (una cadena de 10 caracteres alfanuméricos del tipo DD/MM/AAAA).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Dirección (una cadena de hasta 120 caracteres no vacía).
- Estado civil (se podrá elegir entre soltero, casado, viudo o divorciado)
- Correo electrónico (una cadena de hasta 60 caracteres).
- Formación (una cadena de hasta 200 caracteres).
- Número de cuenta bancaria (una cadena de hasta 20 caracteres).
- Número de la seguridad social (una cadena de hasta 12 números).

→ **RD1.7** Los datos necesarios para buscar un empleado.

- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).

→ **RD1.8** Los datos necesarios para asignar un empleado a una tarea.

- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Nombre de la tarea (una cadena de hasta 50 caracteres no vacía).

→ **RD1.9** Los datos necesarios para modificar una tarea.

- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Nombre de la tarea (una cadena de hasta 50 caracteres no vacía).

→ **RF1.1** Alta de un empleado:

Esta función registra un empleado dentro del sistema.

- Datos de entrada: **RD1.1**.

- Datos almacenados: **RD1.2**.
- Datos de salida: No tiene.

→ **RF1.2** Eliminar un empleado:

Esta función elimina a un empleado del sistema.

- Datos de entrada: **RD1.5**.
- Datos almacenados: **RD1.2**.
- Datos de salida: No tiene.

→ **RF1.3** Modificar un empleado:

Esta función modifica un empleado del sistema.

- Datos de entrada: **RD1.6**.
- Datos almacenados: **RD1.2**.
- Datos de salida: No tiene.

→ **RF1.4** Buscar empleado:

Busca un empleado en el sistema.

- Datos de entrada: **RD1.7**.
- Datos almacenados: **RD1.2**.
- Datos de salida: **RD1.3**.

→ **RF1.5** Asignar tareas:

Asigna un empleado a una tarea.

- Datos de entrada: **RD1.8**.
- Datos almacenados: **RD1.4**.
- Datos de salida: No tiene.

→ **RF1.6** Modificar tareas:

Modifica los empleados que realizan una tarea.

- Datos de entrada: **RD1.9**.
- Datos almacenados: **RD1.4**.
- Datos de salida: No tiene.

→ **RS1.1** Es necesario que el nuevo empleado tenga un mínimo de 16 años para poder añadirlo al sistema.

- Datos a los que referencia: **RD1.1**.
- Requisito funcional al que referencia: **RF1.1**.

→ **RS1.2** Si se elimina un empleado, también se eliminarán sus datos para cada una de las tareas que estaba realizando. **RF1.2**.

- Datos a los que referencia: **RD3.1**.
- Requisito funcional al que referencia: **RF3.1**.

El subsistema de gestión de menús va a tener los siguientes requisitos de datos, requisitos funcionales y restricciones semánticas:

→ **RD2.1** Los datos de entrada de un plato:

- Código de plato (una cadena numérica de 4 dígitos).
- Nombre (una cadena de hasta 25 caracteres no vacía).
- Precio (un número real).
- Ingredientes (una cadena alfanumérica de hasta 50 caracteres no vacía).
- Productos alérgenos (una cadena de hasta 50 caracteres).
- Disponibilidad (una cadena de 2 caracteres).

→ **RD2.2** Los datos de un plato almacenado:

- Código de plato (una cadena numérica de 4 dígitos).
- Nombre (una cadena de hasta 25 caracteres no vacía).
- Precio (un número real).
- ~~Ingredientes (una cadena alfanumérica de hasta 50 caracteres no vacía).~~
- Ingredientes (una lista con los códigos de ingrediente, de hasta 5 caracteres por ingrediente).
- Productos alérgenos (una cadena de hasta 50 caracteres).
- Disponibilidad (una cadena de 2 caracteres).

→ **RD2.3** Los datos de entrada para modificar un plato:

- Código de plato (una cadena numérica de 4 dígitos).
- Nombre (una cadena de hasta 25 caracteres no vacía).
- Precio (un número real).
- ~~Ingredientes (una cadena alfanumérica de hasta 50 caracteres no vacía).~~
- Ingredientes (una lista con los códigos de ingrediente, de hasta 5 caracteres por ingrediente).
- Productos alérgenos (una cadena de hasta 50 caracteres).
- Disponibilidad (una cadena de 2 caracteres).

→ **RD2.4** Los datos de entrada para eliminar un plato:

- Código de plato (una cadena numérica de 4 dígitos).

→ **RD2.5** Los datos de entrada para buscar un plato:

- Código de plato (una cadena numérica de 4 dígitos).

→ **RD2.6** Los datos de salida de un plato:

- Código de plato (una cadena numérica de 4 dígitos).
- Nombre (una cadena de hasta 25 caracteres no vacía).
- Precio (un número real).

- Ingredientes (una cadena alfanumérica de hasta 50 caracteres no vacía).
- Productos alérgenos (una cadena de hasta 50 caracteres).RD3.2.
- Disponibilidad (una cadena de 2 caracteres).

→ **RD2.7** Los datos de entrada de un menú:

- Identificador de menú (cadena numérica de 4 dígitos).
- ~~Código de plato (una cadena numérica de 4 dígitos). Uno por cada plato que compone el menú.~~
- Lista con los códigos de los platos, de hasta ~~50 caracteres~~ 4 caracteres cada uno.
- Precio (un número real).
- Disponibilidad (cadena de hasta 2 caracteres).

→ **RD2.8** Los datos almacenados de un menú:

- Identificador de menú (cadena numérica de 4 dígitos).
- ~~Código de plato (una cadena numérica de 4 dígitos). Uno por cada plato que compone el menú.~~
- Lista con los códigos de los platos, de hasta ~~50 caracteres~~ 4 caracteres cada uno.
- Precio (un número real).
- Disponibilidad (cadena de hasta 2 caracteres).

→ **RD2.9** Los datos de entrada para modificar un menú:

- Identificador de menú (cadena numérica de 4 dígitos).
- ~~Código de plato (una cadena numérica de 4 dígitos). Uno por cada plato que compone el menú.~~
- Lista con los códigos de los platos, de hasta ~~50 caracteres~~ 4 caracteres cada uno.
- Precio (un número real).
- Disponibilidad (cadena de hasta 2 caracteres).

→ **RD2.10** Los datos de entrada para eliminar un menú:

- Identificador de menú (cadena numérica de 4 dígitos).

→ **RF2.1** Añadir un plato.

Esta función registra un nuevo plato en el sistema.

- Datos de entrada: **RD2.1**.
- Datos almacenados: **RD2.2**.
- Datos de salida: No tiene.

→ **RF2.2** Eliminar plato.

Esta función elimina un plato del sistema.

- Datos de entrada: **RD2.4**.
- Datos almacenados: **RD2.2**.
- Datos de salida: No tiene.

→ **RF2.3** Modificar un plato.

Esta función permite modificar la información de un plato almacenado en el sistema.

- Datos de entrada: **RD2.3**.
- Datos almacenados: **RD2.2**.
- Datos de salida: No tiene.

→ **RF2.4** Buscar un plato.

Esta función permite buscar un plato almacenado en el sistema.

- Datos de entrada: **RD2.5**.
- Datos almacenados: **RD2.2**.
- Datos de salida: **RD2.6**.

→ **RF2.5** Añadir un menú.

Esta función registra un nuevo menú en el sistema.

- Datos de entrada: **RD2.7**.
- Datos almacenados: **RD2.8**.
- Datos de salida: No tiene.

→ **RF2.6** Modificar un menú.

Esta función permite modificar la información referente a un menú almacenado en el sistema.

- Datos de entrada: **RD2.9**.
- Datos almacenados: **RD2.8**.
- Datos de salida: No tiene.

→ **RF2.7** Eliminar un menú.

Esta función elimina un menú del sistema.

- Datos de entrada: **RD2.10**.
- Datos almacenados: **RD2.8**.
- Datos de salida: No tiene.

→ **RS2.1** Si se elimina un plato que forma parte de un menú, también se elimina el plato del menú.

- Datos a los que referencia: **RD2.4**.
- Requisito funcional al que referencia: **RF2.2**.

→ **RS2.2** Si se modifica la disponibilidad de un plato a "NO", se elimina ese plato del menú asociado (si tiene).

- Datos a los que referencia: **RD2.3**.
- Requisito funcional al que referencia: **RF2.3**.

El subsistema de gestión de proveedores va a tener la siguiente lista de requisitos de datos, requisitos funcionales y restricciones semánticas:

→ **RD3.1.** Los datos de entrada para incluir a un proveedor:

- Nombre y apellidos (una cadena de hasta 60 caracteres no vacía).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Correo electrónico (una cadena de hasta 60 caracteres).
- Nombre de la empresa en la que trabaja (una cadena de máximo 60 caracteres).
- NIF de la empresa (una cadena alfanumérica de hasta 9 caracteres).
- Tipo de productos que provee (una cadena de 20 caracteres).
- **Edad (entero de 2 dígitos).**

→ **RD3.2.** Los datos almacenados al incluir a un proveedor son los siguientes:

- Nombre y apellido (una cadena de hasta 60 caracteres no vacía).
- DNI (una cadena de hasta 9 caracteres alfanuméricos no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Correo electrónico (una cadena de hasta 60 caracteres).
- Nombre de la empresa en la que trabaja (una cadena de máximo 60 caracteres).
- NIF de la empresa (una cadena alfanumérica de hasta 9 caracteres).
- Tipo de productos que provee (una cadena de 20 caracteres).
- **Edad (entero de 2 dígitos).**

→ **RD3.3.** Los datos de entrada para eliminar a un proveedor son los siguientes:

- DNI del proveedor (una cadena de 9 caracteres alfanuméricos no vacía).

→ **RD3.4.** Los datos de entrada para modificar a un proveedor son:

- DNI del proveedor (una cadena de 9 caracteres alfanuméricos no vacía).
- Datos a modificar, como la empresa para la que trabaja (una cadena de máximo 60 caracteres), el NIF de la empresa (una cadena alfanumérica de hasta 9 caracteres), el nombre del proveedor (una cadena de hasta 60 caracteres no vacía), el teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos), el

correo electrónico (una cadena de hasta 60 caracteres) o el tipo de ingredientes que provee (una cadena de 20 caracteres) **y la edad.**

→ **RD3.5.** Los datos almacenados al modificar a un proveedor son:

- DNI del proveedor (una cadena de 9 caracteres alfanuméricos no vacía).
- Datos que se almacenan son la empresa para la que trabaja (una cadena de máximo 60 caracteres), el NIF de la empresa (una cadena alfanumérica de hasta 9 caracteres), el nombre del proveedor (una cadena de hasta 60 caracteres no vacía), el teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos), el correo electrónico (una cadena de hasta 60 caracteres) o el tipo de ingredientes que provee (una cadena de 20 caracteres).

→ **RD3.6.** Los datos de entrada para buscar a un proveedor son:

- Código de ingrediente (un número entero de 5 dígitos).

→ **RD3.7.** Los datos de salida al buscar a un proveedor son:

- Nombre y apellido (una cadena de hasta 60 caracteres no vacía).
- Teléfono (una cadena de hasta 20 caracteres numéricos en la que el primer carácter puede ser numérico o un signo + y los demás caracteres son numéricos).
- Correo electrónico (una cadena de hasta 60 caracteres).
- Nombre de la empresa en la que trabaja (una cadena de máximo 60 caracteres).
- NIF de la empresa (una cadena alfanumérico de hasta 9 caracteres).
- **Edad (entero de 2 dígitos).**

→ **RD3.8.** Los datos de entrada de los ingredientes son los que siguen:

- Código de ingrediente (un número entero de 5 dígitos).
- El tipo de ingrediente (una cadena de caracteres de longitud máxima 20).
- El nombre (representado con una cadena de 25 caracteres).
- La cantidad (un número entero entre 1 y 500).
- El código del proveedor que lo suministra (cadena de 9 caracteres alfanuméricos).
- **El DNI del proveedor que provee el ingrediente.**

→ **RD3.9.** Los datos que se almacenan de los ingredientes son los que siguen:

- Código de ingrediente (un número entero de 5 dígitos).
- El tipo de ingrediente (una cadena de caracteres de longitud máxima 20).

- El nombre (representado con una cadena de 25 caracteres).
- La cantidad (un número entero entre 1 y 500).
- El DNI del proveedor que provee el ingrediente.

→ **RF3.1.** Incluir un proveedor:

Esta función incluye un nuevo proveedor en el sistema.

- Datos de entrada: **RD3.1.**
- Datos almacenados: **RD3.2.**
- Datos de salida: No tiene.

→ **RF3.2.** Eliminar un proveedor:

Esta función elimina a un proveedor previamente almacenado en el sistema.

- Datos de entrada: **RD3.3.**
- Datos almacenados: **RD3.2.**
- Datos de salida: No tiene.

→ **RF3.3.** Modificar un proveedor:

Esta función los datos almacenados de un proveedor.

- Datos de entrada: **RD3.4.**
- Datos almacenados: ~~RD3.5.~~ **RD3.2.**
- Datos de salida: No tiene.

→ **RF3.4.** Buscar un proveedor:

Esta función permite buscar a un proveedor concreto dentro del sistema.

- Datos de entrada: **RD3.6.**
- Datos almacenados: **RD3.2.**
- Datos de salida: **RD3.7.**

→ **RF3.5.** Establecer ingrediente provisto:

Esta función establece el ingrediente provisto por un proveedor.

- Datos de entrada: **RD3.8.**
- Datos almacenados: **RD3.9.**
- Datos de salida: No tiene.

→ **RS3.1.** La persona debe ser mayor de 18 años para poder apuntarse en el sistema como proveedor.

- Datos a los que referencia: **RD3.1.**
- Requisito funcional al que referencia: **RF3.1.**

Validación cruzada de requisitos

RD	Entrada	Manejo	Salida
RD1.1	RF1.1		
RD1.2		RF1.1, RF1.2, RF1.3, RF1.4	
RD1.3			RF1.4
RD1.4		RF1.5, RF1.6	
RD1.5	RF1.2		
RD1.6	RF1.3		
RD1.7	RF1.4		
RD1.8	RF1.5		
RD1.9	RF1.6		
RD2.1	RF2.1		
RD2.2		RF2.1, RF2.2, RF2.3, RF2.4	
RD2.3	RF2.3		
RD2.4	RF2.2		
RD2.5	RF2.4		
RD2.6			RF2.4
RD2.7	RF2.5		
RD2.8		RF2.5, RF2.6, RF2.7	
RD2.9	RF2.6		
RD2.10	RF2.7		
RD3.1	RF3.1		
RD3.2		RF3.1, RF3.4, RF3.3	
RD3.3	RF3.2		
RD3.4	RF3.3		
RD3.5		RF3.3	
RD3.6	RF3.4		
RD3.7			RF3.4
RD3.8	RF3.5		
RD3.9		RF3.5	

RF	Entrada	Manejo	Salida
RF1.1	RD1.1	RD1.2	
RF1.2	RD1.5	RD1.2	
RF1.3	RD1.6	RD1.2	
RF1.4	RD1.7	RD1.2	RD1.3
RF1.5	RD1.8	RD1.4	
RF1.6	RD1.9	RD1.4	
RF2.1	RD2.1	RD2.2	
RF2.2	RD2.4	RD2.2	
RF2.3	RD2.3	RD2.2	
RF2.4	RD2.5	RD2.2	RD2.6
RF2.5	RD2.7	RD2.8	
RF2.6	RD2.9	RD2.8	
RF2.7	RD2.10	RD2.8	
RF3.1	RD3.1	RD3.2	
RF3.2	RD3.3	RD3.2	
RF3.3	RD3.4	RD3.5 RD3.2	
RF3.4	RD3.6	RD3.2	RD3.7
RF3.5	RD3.8	RD3.9	

PRÁCTICA 2

Modificaciones de prácticas anteriores

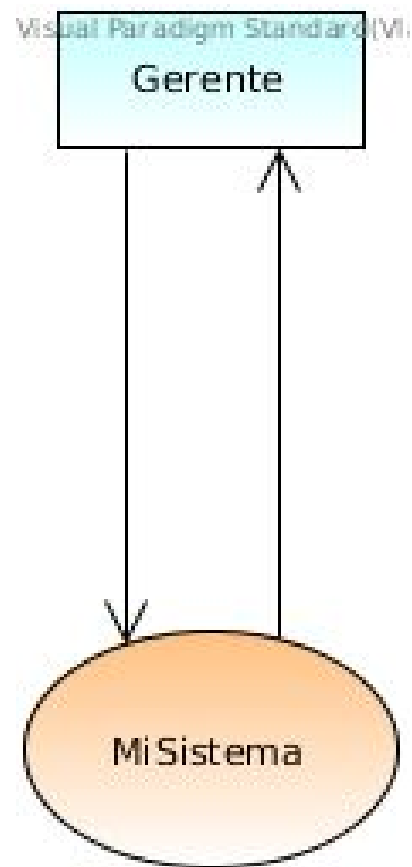
- Se ha eliminado el requisito de datos 3.5 debido a que es el mismo que el 3.2.
- Se han modificado los requisitos de datos 2.8 y 2.9, haciendo que se guarde una lista con los códigos de platos.
- Se han modificado los requisitos de datos 3.8 y 3.9, haciendo que se guarde el código del proveedor que ha provisto dicho ingrediente al restaurante.

Desarrollo de la práctica

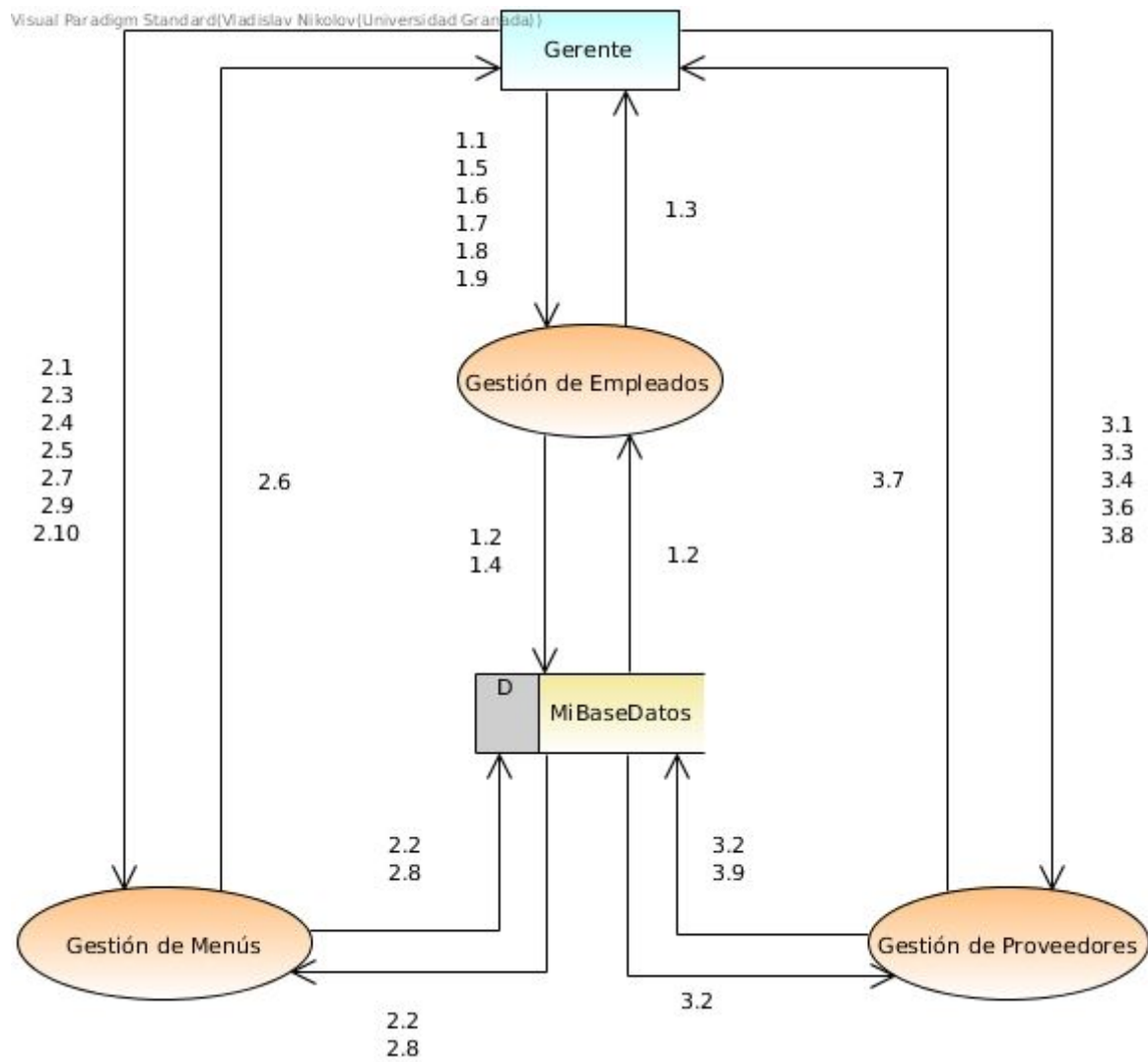
El trabajo se ha dividido de la siguiente forma:

- ➔ Gestión de empleados: José María Sánchez Guerrero
- ➔ Gestión de menús: Vladislav Nikolov Vasilev
- ➔ Gestión de proveedores: Nazaret Román Guerrero

Esquema de caja negra

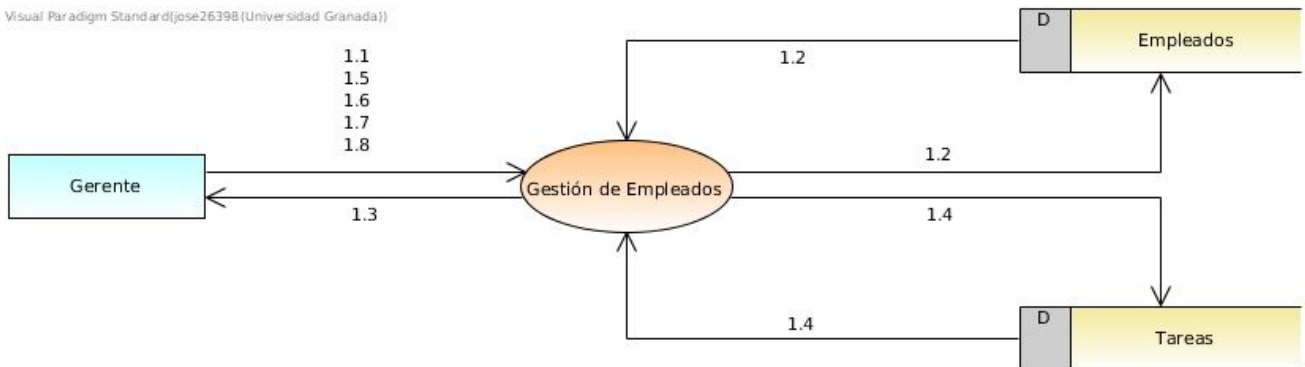


Esquema almacén

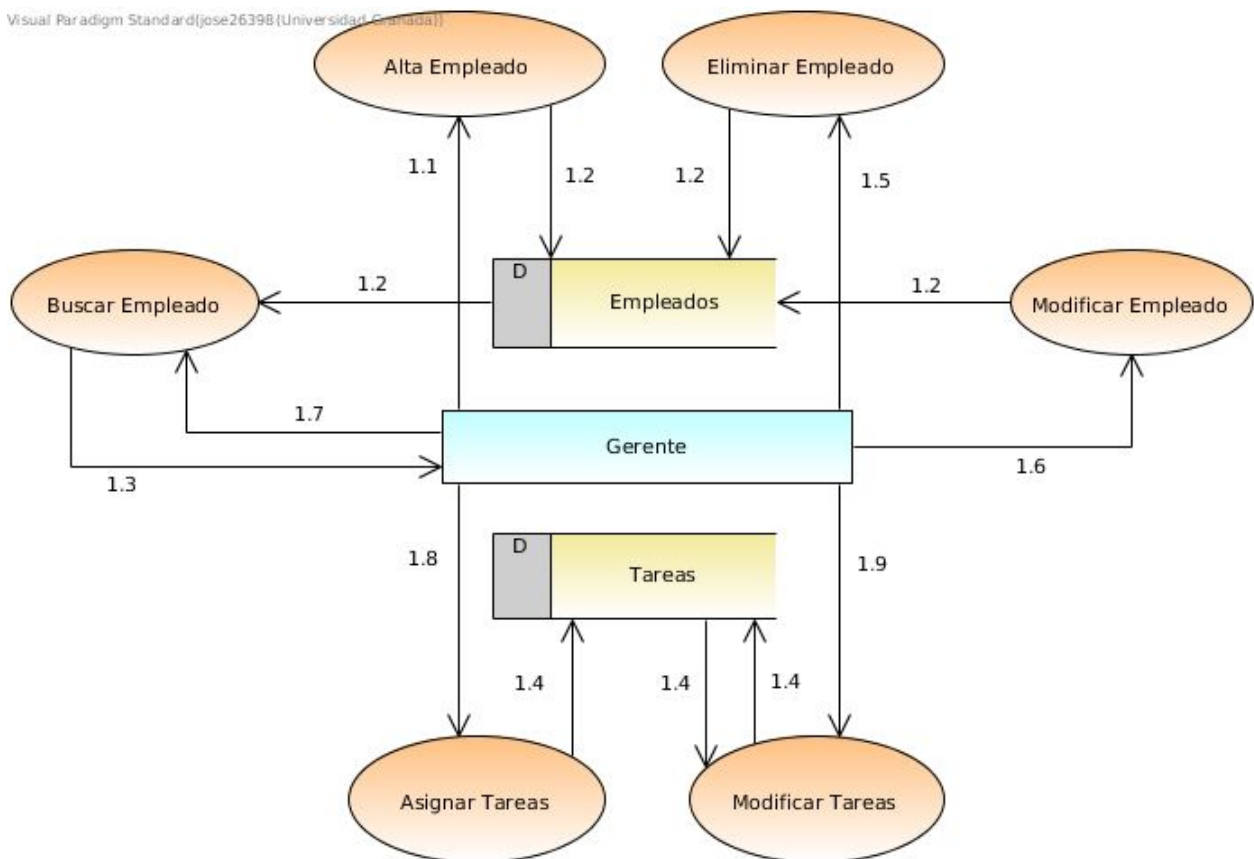


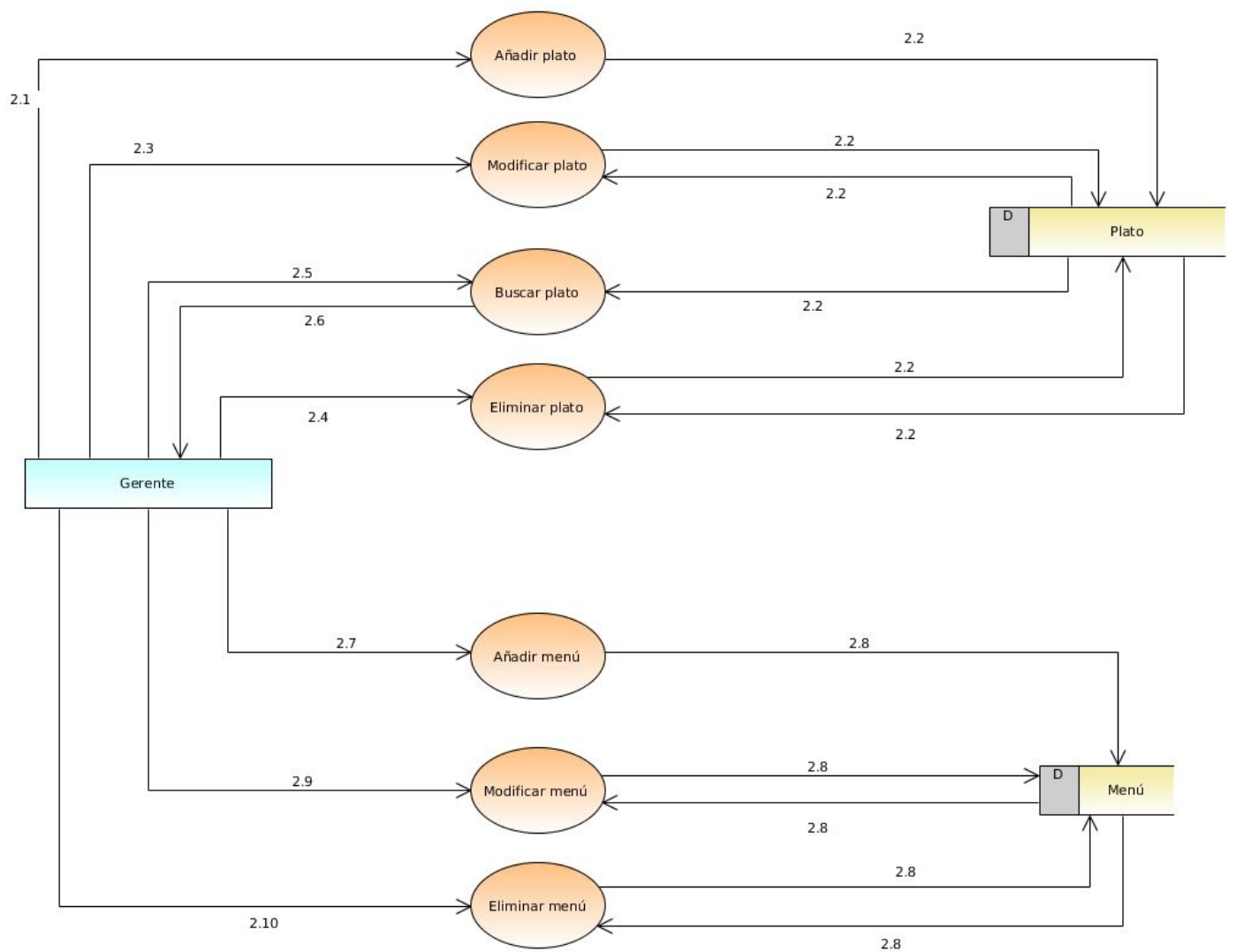
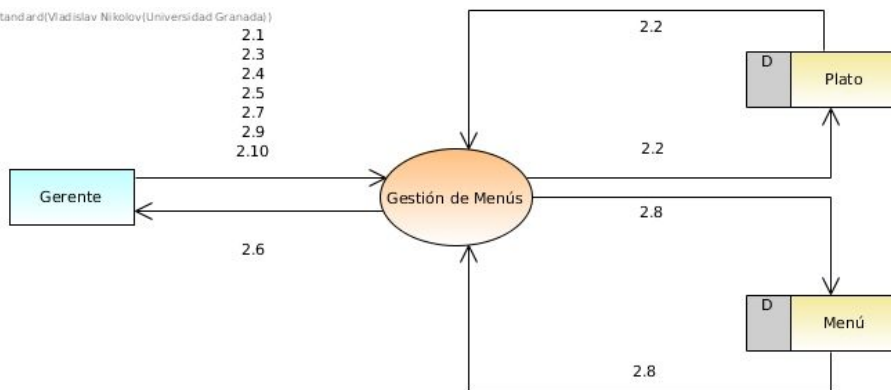
Refinamientos de cada subsistema

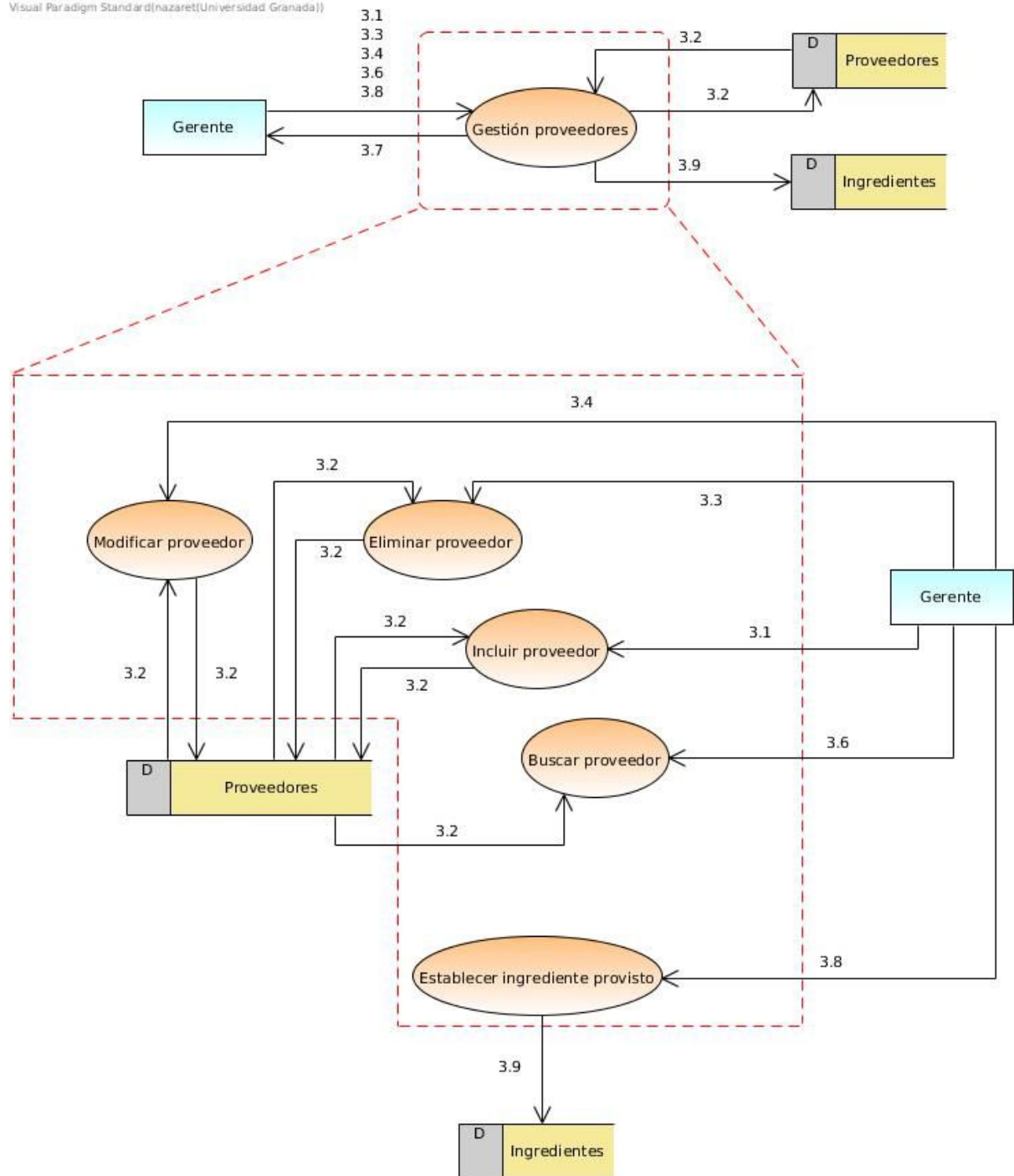
Visual Paradigm Standard(jose26398(Universidad Granada))



Visual Paradigm Standard(jose26398(Universidad Granada))

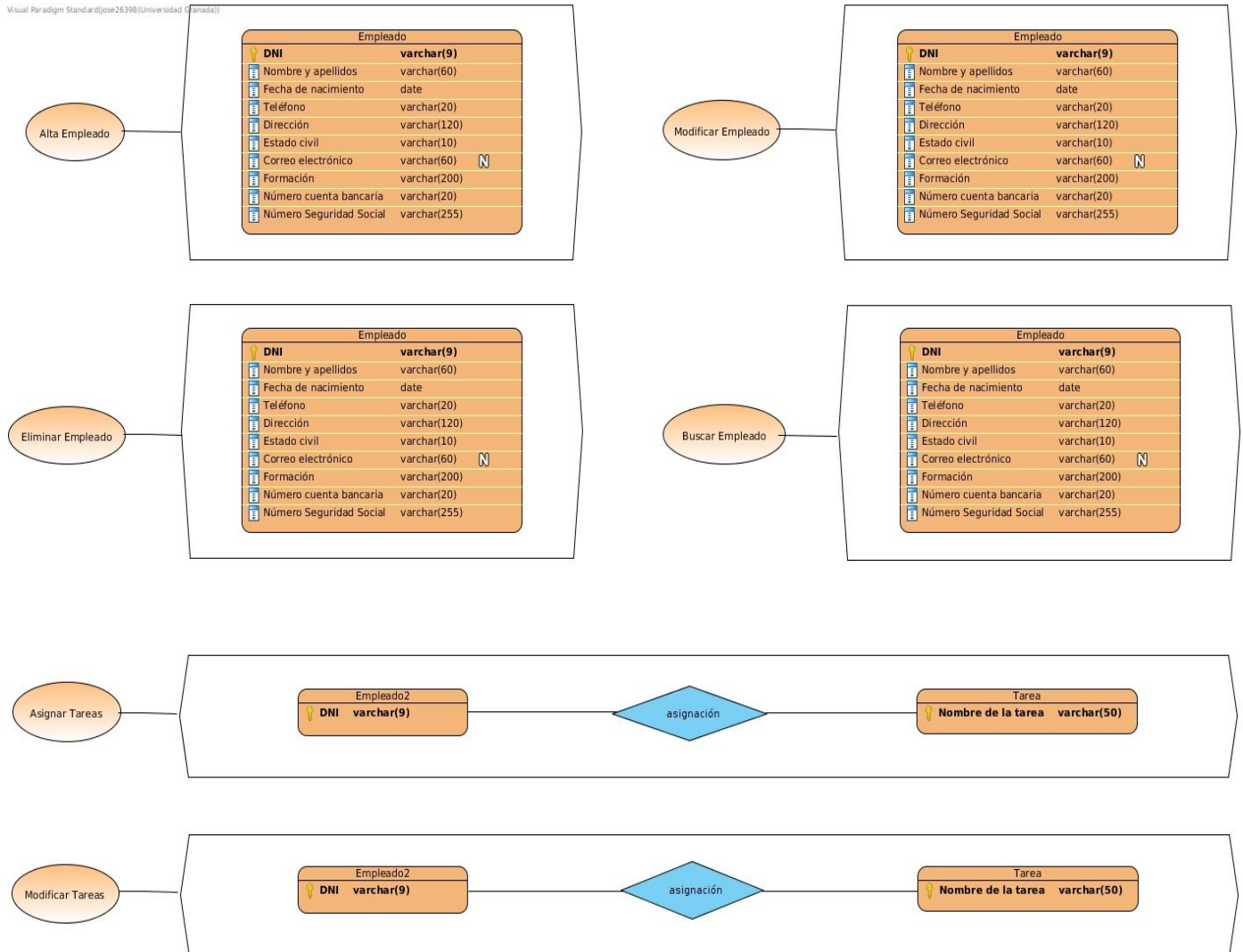


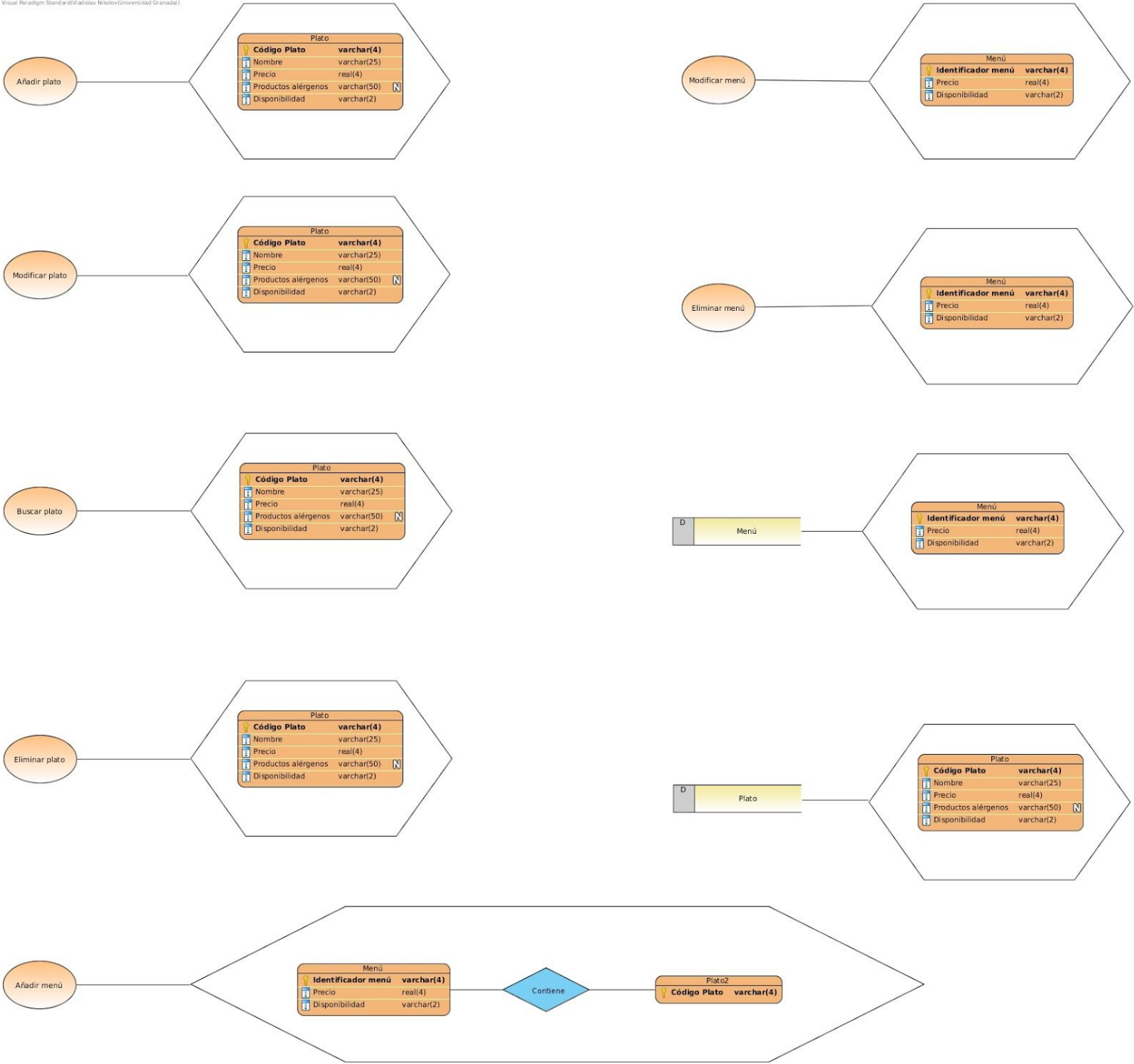


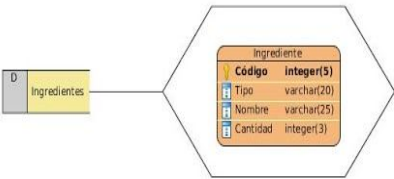
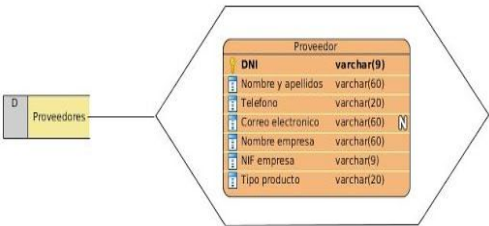
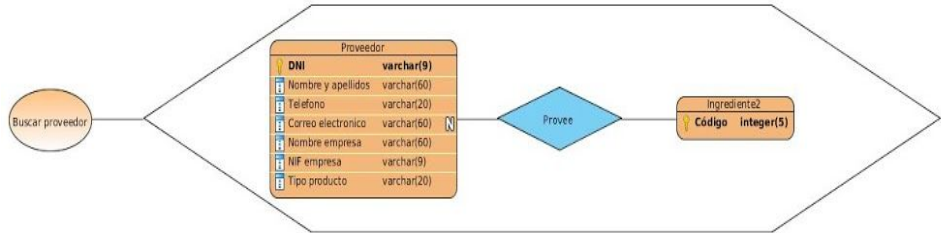
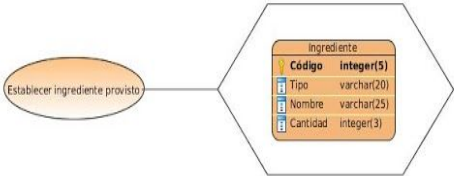
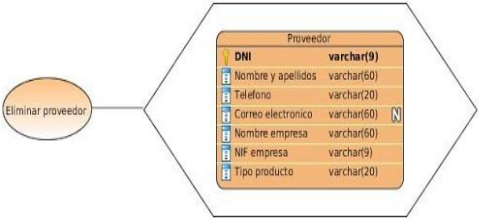
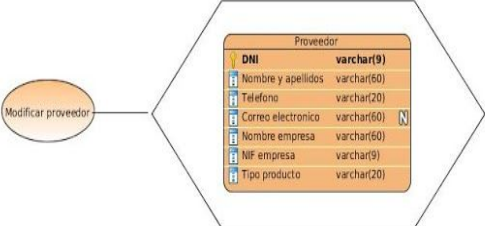
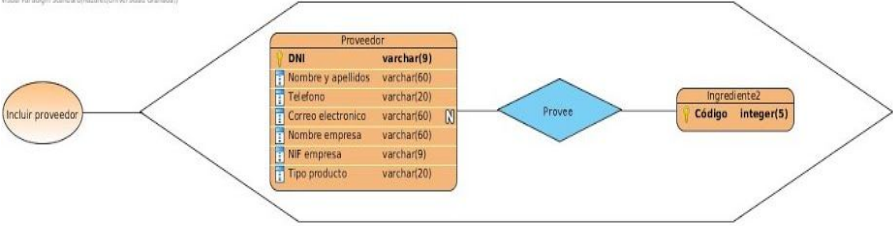


Esquemas externos del refinamiento

Visual Paradigm Standard(jose26398(Universidad C...







Esquemas externos de los subsistemas

Visual Paradigm Standard(Vladislav Nikolov(Universidad Granada))

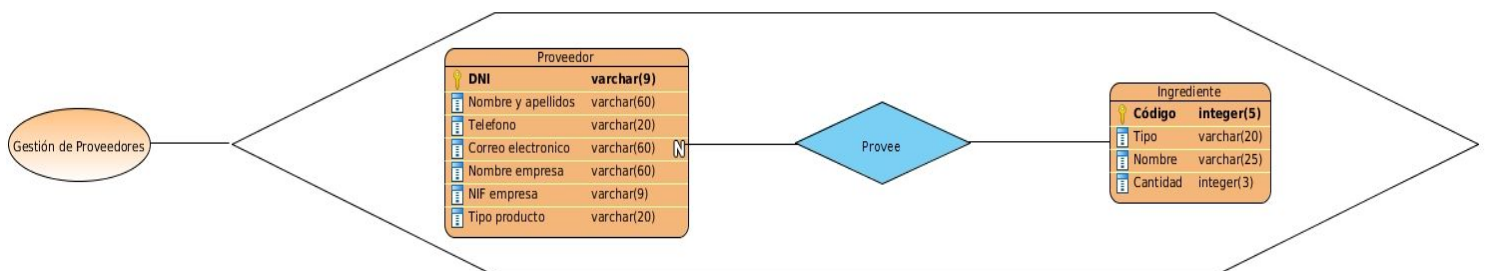
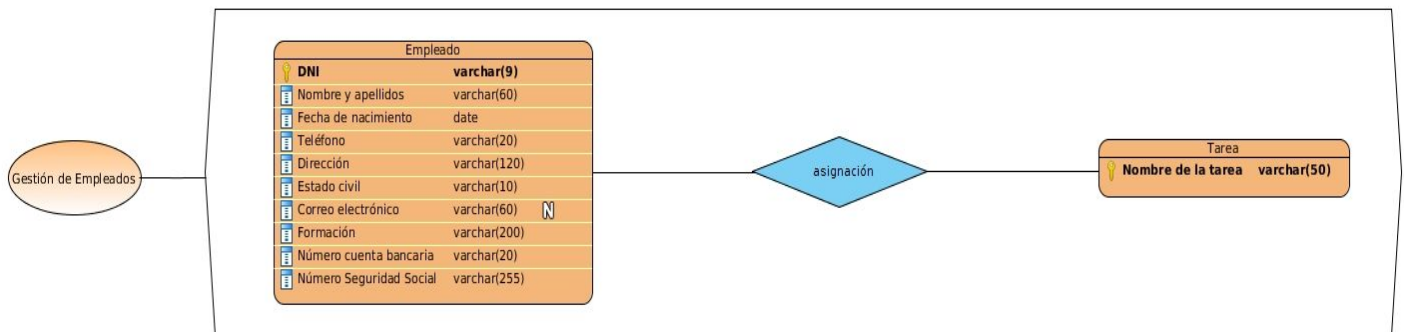
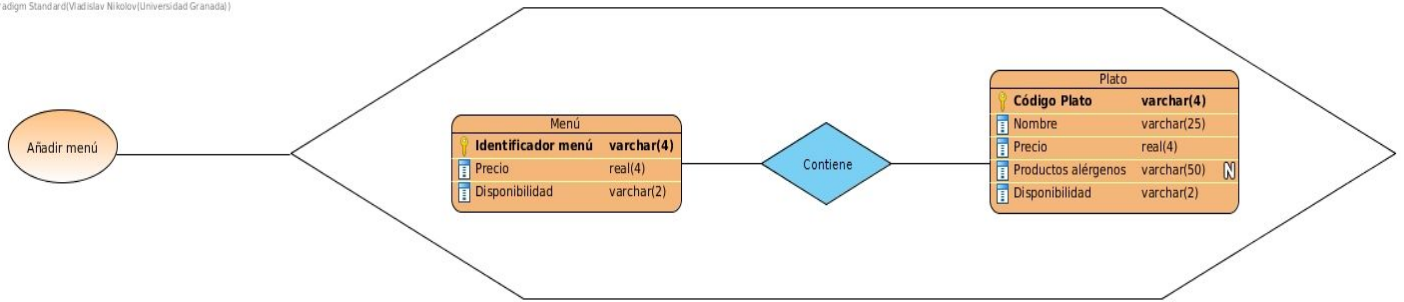
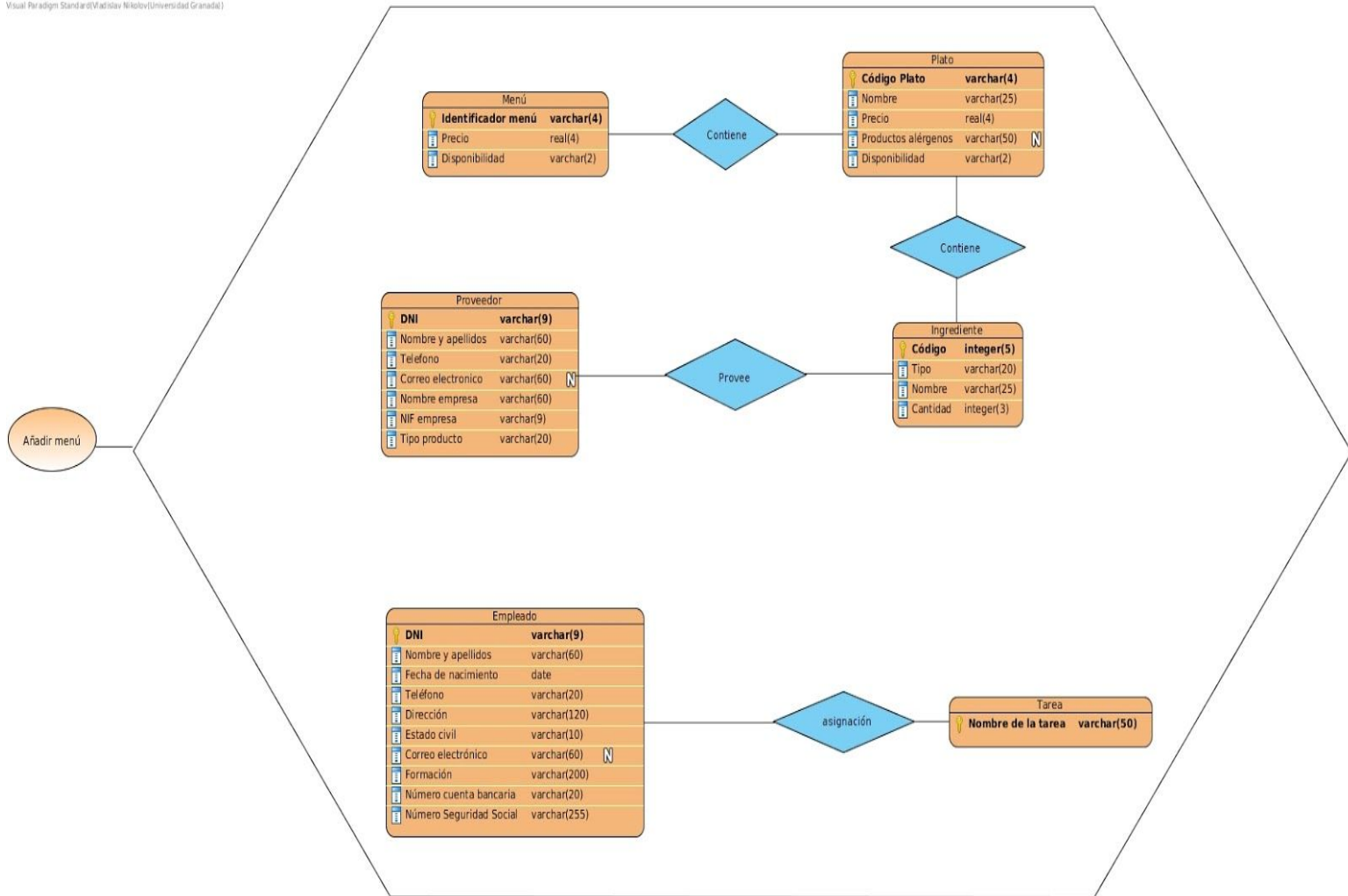


Diagrama E/R final

Visual Paradigm Standard(Vladislav Nikolov(Universidad Granada))



PRÁCTICA 3

Modificaciones de prácticas anteriores

- Se han corregido los diagramas correspondientes a los esquemas externos de los subsistemas y el diagrama E/R final, añadiendo la clave primaria a la entidad Tarea (ahora tiene una clave primaria que es **Nombre de la tarea**).
- Se ha añadido un dato a los proveedores (la edad del proveedor).

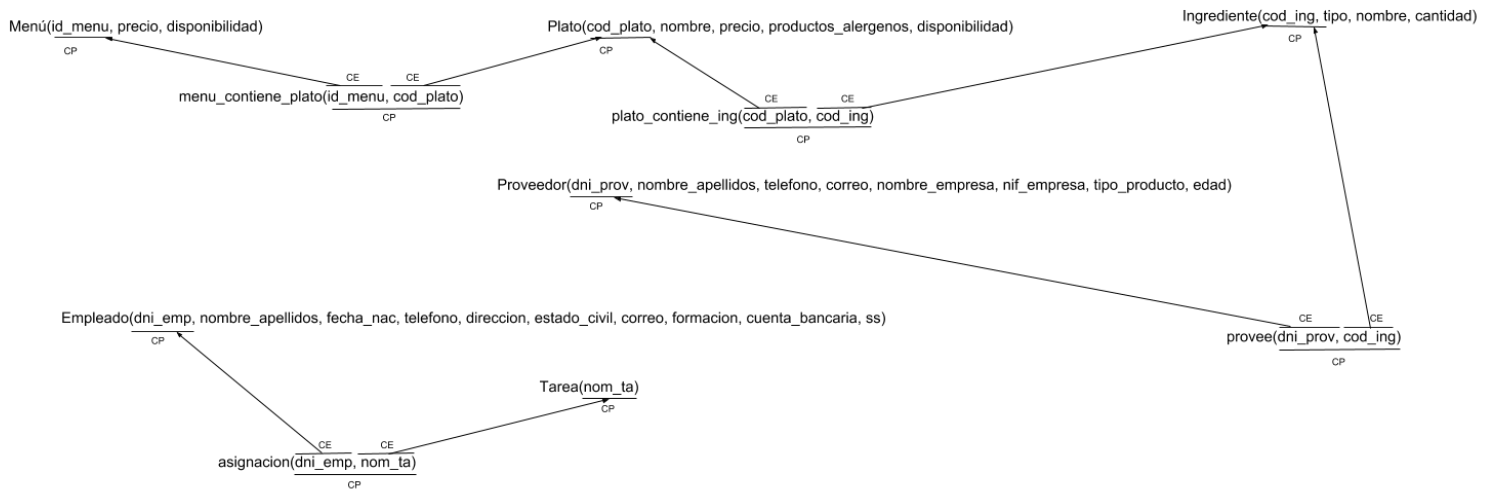
Desarrollo de la práctica

El trabajo se ha dividido de la siguiente forma:

- Gestión de empleados: José María Sánchez Guerrero
- Gestión de menús: Vladislav Nikolov Vasilev
- Gestión de proveedores: Nazaret Román Guerrero

Se ha elegido MySQL como Sistema de Gestión de Base de Datos. Por tanto, todas las sentencias SQL usarán la sintaxis y los tipos de datos de MySQL.

Paso a tablas



Diseño físico

-- Subsistema Gestión de Empleados

```
CREATE TABLE Empleado (
    dni_emp VARCHAR(9) PRIMARY KEY,
    nombre_apellidos VARCHAR(60) NOT NULL,
    fecha_nac DATE NOT NULL,
    telefono VARCHAR(20) NOT NULL,
    direccion VARCHAR(120) NOT NULL,
    estadocivil VARCHAR(10) NOT NULL
    CHECK (estadocivil IN ('soltero', 'casado', 'divorciado', 'viudo')),
    correo VARCHAR(60),
    formacion VARCHAR(200) NOT NULL,
    cuenta_bancaria VARCHAR(20) NOT NULL,
    ss VARCHAR(12) NOT NULL
);
```

```
CREATE TABLE Tarea(
    nom_ta VARCHAR(50) PRIMARY KEY
);
```

-- Subsistema Gestión de Menús

```
CREATE TABLE Menu (  
    id_menu VARCHAR(4) PRIMARY KEY,  
    precio FLOAT(5,2) NOT NULL CHECK (precio >= 0),  
    disponibilidad VARCHAR(2) NOT NULL CHECK (disponibilidad IN ('SI', 'NO'))  
);
```

```
CREATE TABLE Plato (  
    cod_plato VARCHAR(4) PRIMARY KEY,  
    nombre VARCHAR(25) NOT NULL,  
    precio FLOAT(5,2) NOT NULL CHECK (precio >= 0),  
    productos_alergenos VARCHAR(50) NOT NULL,  
    disponibilidad VARCHAR(2) NOT NULL CHECK (disponibilidad IN ('SI', 'NO'))  
);
```

-- Subsistema Gestión de Proveedores

```
CREATE TABLE Proveedor(  
    dni varchar(9) PRIMARY KEY,  
    nombre_apellidos varchar(60) NOT NULL,  
    telefono varchar(20) NOT NULL,  
    correo varchar(60),  
    nombre_empresa varchar(60) NOT NULL,  
    nif_empresa varchar(9) NOT NULL,  
    tipo_producto varchar(20) NOT NULL,  
    edad INT NOT NULL  
);
```

```
CREATE TABLE Ingrediente(  
    cod_ing varchar(5) PRIMARY KEY,  
    tipo varchar(20) NOT NULL,  
    nombre varchar(25) NOT NULL,  
    cantidad integer(3) NOT NULL CHECK(1 <= cantidad <= 500)  
);
```

-- Tablas de relaciones

-- Subsistema Gestión de Empleados

```
CREATE TABLE asignacion(  
    dni_emp VARCHAR(9) REFERENCES Empleado(dni_emp),  
    nom_ta VARCHAR(50) REFERENCES Tarea(nom_ta),  
    PRIMARY KEY (dni_emp, nom_ta)  
);
```

-- Subsistema Gestión de Menús

```
CREATE TABLE menu_contiene_plato (  
    id_menu VARCHAR(4) REFERENCES Menu(id_menu),  
    cod_plato VARCHAR(4) REFERENCES Plato(cod_plato),  
    PRIMARY KEY(id_menu, cod_plato)  
);
```

```
CREATE TABLE plato_contiene_ing (  
    cod_plato VARCHAR(4) REFERENCES Plato(cod_plato),  
    cod_ing varchar(5) REFERENCES Ingrediente(cod_ing),  
    PRIMARY KEY(cod_plato, cod_ing)  
);
```

-- Subsistema Gestión de Proveedores

```
CREATE TABLE provee(  
    dni_prov varchar(9) REFERENCES Proveedor(dni_prov),  
    cod_ing varchar(5) REFERENCES Ingrediente(cod_ing),  
    PRIMARY KEY(dni_prov, cod_ing)  
);
```

Disparadores

USE pomodoro;

-- Gestión de Empleados

DROP TRIGGER IF EXISTS eliminar_emp;

```
DELIMITER |  
CREATE TRIGGER eliminar_emp BEFORE DELETE ON Empleado  
    FOR EACH ROW  
    BEGIN  
        DELETE FROM asignacion WHERE asignacion.dni_emp = OLD.dni_emp;  
    END;  
|  
DELIMITER ;
```

-- Gestión de Menús

DROP TRIGGER IF EXISTS eliminar_plato;

```
DELIMITER |
```

```
CREATE TRIGGER eliminar_plato BEFORE DELETE ON Plato
FOR EACH ROW
BEGIN
    DELETE FROM menu_contiene_plato WHERE menu_contiene_plato.cod_plato =
    OLD.cod_plato;
END;
|
DELIMITER ;
```

-- Gestión de Proveedores

```
DROP TRIGGER IF EXISTS disp_proveedores;
```

```
DELIMITER |
CREATE TRIGGER disp_proveedores
BEFORE INSERT ON Proveedor FOR EACH ROW
BEGIN
    IF NEW.edad < 18 THEN
        SIGNAL SQLSTATE '55555'
        SET MESSAGE_TEXT = 'Error en la edad del proveedor. No puede ser
menor que 18';
    END IF;
END;
|
DELIMITER ;
```

PRÁCTICA 4

Modificaciones de prácticas anteriores

- Se han modificado los Requisitos de Datos RD2.1, RD2.2, RD2.3, RD2.7, RD2.8 y RD2.9. Las modificaciones se han puesto en color **naranja**. La información anterior se ha tachado.

Desarrollo de la práctica

El trabajo se ha dividido de la siguiente forma:

- Gestión de empleados: José María Sánchez Guerrero
- Gestión de menús: Vladislav Nikolov Vasilev
- Gestión de proveedores: Nazaret Román Guerrero

Se ha realizado una implementación del sistema completo utilizando el lenguaje de programación Java (versión 8) y el IDE Netbeans. Para poder ofrecer una interfaz gráfica al usuario, se ha utilizado el paquete Swing de Java.

Adicionalmente a esto se han añadido los *triggers* necesarios para cumplir las restricciones semánticas del sistema, y algunos más para asegurarse de que MySQL comprase ciertas restricciones sobre los datos al insertarlos. Se han añadido también una serie de datos iniciales que tiene la base de datos para poder hacer pruebas sobre estos.