

**BUAP**

**FCC**

**INGENIERIA DE  
SOFTWARE**

**PROFESOR: JUDITH  
PEREZ MARCIAL**



**ALUMNOS:  
PERALTA MARTINEZ  
JOSE RAFAEL**

**VELAZCO NOLASCO  
KEVIN**

**CASTRO GONZALES  
ABIGAIL**

**RAMIREZ SANLUIS  
VICTOR SEMEY**

**PRIMAVERA 2024**

**2 DE MAYO DE 2024**

### **Revisión de los requisitos del proyecto:**

Se comenzó por revisar cuidadosamente los requisitos del proyecto proporcionados en la descripción. Esto nos permitió comprender completamente qué funcionalidades debe soportar el sistema y qué datos debe gestionar.

### **Identificación de las entidades principales:**

Basándonos en los requisitos, identificamos las entidades principales del sistema. Estas entidades generalmente corresponden a objetos del mundo real o conceptos clave dentro del dominio del problema. Por ejemplo, en un sistema de gestión de consultorios médicos, las entidades principales podrían ser Paciente, Médico, Cita, Historia Clínica, etc.

### **Definición de las clases:**

Cada entidad identificada se convirtió en una clase en el diagrama de clases. Para cada clase, definimos sus atributos, que representan las características o datos asociados con esa entidad. Por ejemplo, la clase Paciente podría tener atributos como nombre, dirección, fecha de nacimiento, etc.

### **Identificación de métodos:**

Además de los atributos, también identificamos los métodos que cada clase debería tener. Estos métodos representan el comportamiento de la clase y las acciones que puede realizar. Por ejemplo, la clase Cita podría tener métodos para reservar una cita, cancelar una cita, etc.

### **Determinación de las relaciones entre clases:**

Después de definir las clases y sus atributos y métodos, identificamos las relaciones entre las clases. Estas relaciones modelan cómo interactúan las diferentes entidades dentro del sistema. Las relaciones pueden ser de varios tipos, como asociaciones, agregaciones, composiciones o herencias.

### **Refinamiento del diseño:**

Una vez que se establecieron las clases y las relaciones entre ellas, refinamos el diseño del diagrama de clases para asegurarnos de que fuera claro, coherente y cumpliera con los requisitos del proyecto. Esto incluyó la eliminación de redundancias, la simplificación de relaciones complejas y la revisión de nombres de clases y atributos para mayor claridad.

### **Atributos y métodos de la clase Paciente:**

Se incluyeron atributos como nombre, dirección, fecha de nacimiento, etc., porque son datos básicos necesarios para identificar y gestionar la información de los pacientes en el sistema. Los métodos asociados podrían incluir funciones para

acceder y modificar estos atributos, así como para realizar acciones relacionadas con los pacientes, como actualizar su información personal.

#### **Atributos y métodos de la clase Médico:**

Se incluyeron atributos como nombre y especialidad para identificar y describir a los médicos que trabajan en el consultorio. Los métodos asociados podrían incluir funciones para acceder y modificar estos atributos, así como para realizar acciones específicas relacionadas con los médicos, como reservar y cancelar citas para ellos.

#### **Atributos y métodos de la clase Cita:**

Se incluyeron atributos como fecha, hora y estado para representar la información asociada con cada cita programada en el sistema. Los métodos asociados podrían incluir funciones para reservar y cancelar citas, así como para actualizar su estado según sea necesario.

#### **Atributos y métodos de otras clases:**

Los atributos y métodos de otras clases, como Historia Clínica, Factura y Receta Electrónica, se seleccionaron de manera similar para representar la información y las funcionalidades asociadas con cada entidad dentro del sistema.

#### **Relación entre Cita y Paciente:**

La clase Cita tiene una relación con la clase Paciente. Esta relación indica que una cita está asociada con un paciente en particular. Un paciente puede tener múltiples citas a lo largo del tiempo, pero una cita está vinculada a un único paciente.

#### **Relación entre Cita y Medico:**

La clase Cita también tiene una relación con la clase Medico. Esto indica que una cita está asociada con un médico en particular. Un médico puede tener múltiples citas programadas con diferentes pacientes, pero una cita está vinculada a un solo médico.

#### **Relación entre Factura y Paciente:**

La clase Factura tiene una relación con la clase Paciente. Esta relación indica que una factura está asociada con un paciente en particular. Un paciente puede tener múltiples facturas generadas por los servicios médicos recibidos en el consultorio.

#### **Relación entre HistoriaClínica y Paciente:**

La clase HistoriaClínica tiene una relación con la clase Paciente. Esto indica que la historia clínica de un paciente está asociada con ese paciente específico. Cada paciente tiene su propia historia clínica que registra su historial médico a lo largo del tiempo.

#### **Relación entre RecetaElectrónica y Paciente:**

La clase RecetaElectrónica tiene una relación con la clase Paciente. Esto indica que una receta electrónica está asociada con un paciente en particular. Una receta electrónica puede contener medicamentos prescritos para un paciente específico.

#### **Relación entre RecordatorioSeguimiento y Usuario:**

La clase RecordatorioSeguimiento tiene una relación con la clase Usuario. Esto indica que un recordatorio de seguimiento puede ser utilizado por uno o más usuarios del sistema. Un usuario puede tener varios recordatorios de seguimiento asociados con su cuenta.

#### **Herencia entre Usuario y Administrador:**

La clase Usuario tiene una relación de herencia con la clase Administrador. Esto indica que la clase Administrador hereda atributos y métodos de la clase Usuario. Un administrador es un tipo especial de usuario que tiene acceso adicional y privilegios dentro del sistema.



