

# UACM



Diseño de software

Vista de patrones

SAR

Valadez Carmona Guadalupe Yamileth

Rodríguez Cervantes Kevin Manzur

Cruz Ovando Cristela Adelaida

## HISTORIAL DE VERSIONES

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR@S
30/01/2025	0.50	¶ Versión preliminar del análisis de los puntos de vista.	Guadalupe Yamileth, Manzur Rodríguez, Cristela Adelaida
31/01/2025	1.50	¶ Nuevo formato del documento.	Guadalupe Yamileth, Manzur Rodríguez, Cristela Adelaida
21/02/2025	2.00	¶ Actualización del documento, siguiente el 'Estándar de Documentación V - 2.00'	Manzur Rodríguez
07/03/2025	2.10	¶ 5.1 Introducción. ¶ 5.2 Punto de vista contextual.	Guadalupe Yamileth
14/03/2025	2.20	¶ 5.3 Punto de vista de la composición.	Manzur Rodríguez
23/03/2025	2.30	¶ 5.4 Punto de vista lógico.	Cristela Adelaida
23/03/2025	2.40	¶ 5.5 Punto de vista de la dependencia.	Guadalupe Yamileth
23/03/2025	2.50	¶ 5.6 Punto de vista informativo.	Manzur Rodríguez
11/04/2025	2.60	¶ 5.7 Punto de vista de uso de patrones.	Cristela Adelaida

## INDICE

1. Introducción .....	1
2. Punto de vista contextual .....	1
2.1. Caso de usos detallado.....	1
2.1.1 Comunidad -> Mostrar .....	1
2.1.2 SAR -> Comprobar .....	2
2.1.3 SAR -> Permitir / Denegar .....	3
2.1.4 SAR -> Leer .....	4
2.1.5 Vigilante -> Imprimir Acceso Temporal .....	4
2.1.6 Vigilante -> Registrar .....	5
2.1.7 Visitante -> Validación Acceso Temporal .....	6
2.2. Diagrama de Contexto .....	8
3. Punto de vista de la composición .....	8
3.1. Diagrama de paquetes.....	8
3.2. Diagrama de componentes .....	9
3.3. IDEF0 .....	10
3.3.1 Sistema actual .....	10
3.3.2 Nuevo sistema .....	10
3.4. Diagrama GANTT .....	11
3.4.1 Estimación de Tiempo .....	12
3.5. Entidad de diseño.....	12
3.5.1 Patrón .....	12
3.5.2 Framework.....	12
3.6. Diagrama HIPO .....	12
4. Punto de vista lógico .....	13
4.1. Propósito.....	13
4.2. Problemas de diseño .....	13
4.3. Elementos de diseño .....	14
4.3.1 Entidades de diseño .....	14
4.3.2 Relaciones de diseño .....	14
4.3.3 Atributos de diseño.....	14
4.3.4 Restricciones de diseño.....	14
4.4. Ejemplos de idiomas.....	15

4.5. Diagrama de clases UML .....	15
4.6. Diagrama de objeto UML .....	15
5. Punto de vista de la dependencia.....	16
5.1. Problemas de diseño .....	16
5.1.1 Elementos de diseño .....	16
5.1.2 Atributos de dependencias.....	17
5.2. Ejemplos de Idiomas .....	17
6. Punto de vista de uso de patrones .....	18
6.1.1 Problemas de diseño .....	19
6.1.2 Elementos de diseño .....	19
6.1.3 Ejemplos de idiomas.....	19
6.2. Diagrama de Estructura Compuesta UML .....	19
7. Definiciones, acrónimos y abreviaturas .....	20
8. Bibliografía .....	21

## 1. Introducción

Se detalla los diferentes puntos de vista del proyecto Sistema de Acceso Rápido (S.A.R.). Los diagramas utilizados para representar los diferentes puntos son UML.

## 2. Punto de vista contextual

Sistema y actores que interactúan con el sistema, o las acciones a realizar por parte S.A.R.

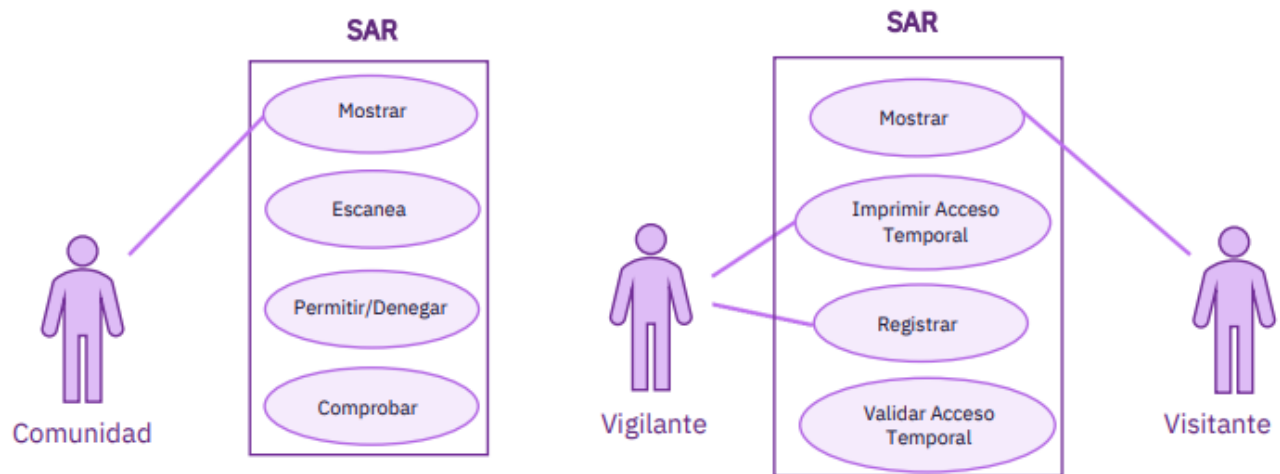


Diagrama de casos de uso

Guadalupe Yamileth  
Diagrama 1

2 horas

### 2.1. Caso de usos detallado.

2.1.1 Comunidad -> Mostrar

Nombre de caso de Uso:	Comunidad -> Mostrar
Actor principal:	Comunidad
Precondiciones:	<ul style="list-style-type: none"> <li>› La comunidad debe mostrar su código QR (ubicado en su credencial) al lector de QR.</li> <li>› Si no está disponible el lector instalado fijamente en la entrada, el vigilante realizará la lectura del QR, con un lector portátil.</li> </ul>
Postcondiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› El código QR debe pertenecer a la UACM.</li> <li>› El código QR debe ser visible completamente.</li> <li>› La credencial debe encontrarse en buenas condiciones, para visualizar la información.</li> </ul>

Escenario Principal:	Acceso 1
Excepciones o Flujos Alternativos:	<ul style="list-style-type: none"> <li>› El usuario debe portar su credencial de identificación perteneciente a la UACM.</li> <li>› Constancia de inscripción proporcionada por la UACM, el cual contenga su código QR.</li> </ul>
Requisitos especiales:	El código QR debe ser visible para el escáner.
Frecuencia:	Alta, la comunidad mostrará su código QR cada vez que dese acceder al plantel.
Temas Abiertos:	<ul style="list-style-type: none"> <li>› Se mostrará la información más relevante del estudiante (nombre, carrera y una foto del mismo para corroborar que es la persona que dice ser.</li> <li>› En caso de que el QR no pertenezca a un alumno, aparecerá un mensaje de error</li> </ul>

### 2.1.2 SAR -> Comprobar


Comprobar	
Nombre de caso de Uso:	SAR -> Comprobar
Actor principal:	Comunidad
Precondiciones:	<ol style="list-style-type: none"> <li>1. Un usuario, ya mostró su código QR</li> <li>2. El escáner extrae el identificador dentro de la URL.</li> <li>3. Con las características del identificador, se determinara que DB realizara la comprobación</li> <li>4. Retornar una respuesta.</li> </ol>
Postcondiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› La información del QR debe ser enviada en texto plano.</li> <li>› La información del QR no excederá un tamaño de 250 caracteres.</li> <li>› El área de sistemas nos indicará que tablas de la DB, contienen la información necesaria para determinar si el usuario que desea entrar pertenece a la comunidad.</li> <li>› El área de sistemas nos indicará que tablas de la DB, contienen la información de los visitantes, donde se almacena la duración de los QR.</li> <li>› Contar con un usuario con permisos, para realizar las consultas a la DB.</li> </ul>
Escenario Principal:	<ul style="list-style-type: none"> <li>› Funcionamiento dentro de la clase “Usuario”.</li> <li>› La cual compruebe si la información enviada por el lector cumple ciertas condiciones, las cuales</li> </ul>

	nos permitirán determinar si se realizó una lectura del código QR de forma correcta.
Excepciones o Flujos Alternativos:	<ul style="list-style-type: none"> <li>› Si el QR, no pertenece a la UACM, denegara el acceso.</li> <li>› Si la información recibida no cumple cierta condición, indicara que es necesario volver a leer el QR.</li> </ul>
Requisitos especiales:	El QR guarna una dirección web, la cual contiene la matricula del usuario.
Frecuencia:	<ul style="list-style-type: none"> <li>› Alta, se van a realizarán un número elevado de consultas por día.</li> <li>› En ciertas horas, aumenta el número de consultas.</li> </ul>
Temas Abiertos:	<ul style="list-style-type: none"> <li>› Como se retornará el resultado de la consulta, la cual determina si pertenece a la UACM.</li> <li>› Que pasa, si la DB, no se encuentra disponible o está saturada.</li> </ul>

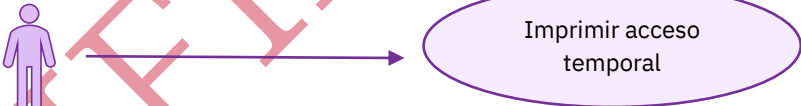
### 2.1.3 SAR -> Permitir / Denegar

	Permitir/Denegar
Nombre de caso de Uso:	SAR -> Permitir / Denegar
Actor principal:	SAR
Precondiciones:	Respuesta de la consulta realizada a la DB.
Post condiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› Un usuario de la comunidad deberá estar activo en el sistema para que se le pueda permitir el acceso, de lo contrario se denegará la entrada.</li> <li>› Si es un visitante, su QR debe encontrarse activo.</li> </ul>
Escenario Principal:	Vista utilizada en la entrada principal.
Excepciones o Flujos Alternativos:	Si la respuesta de la consulta presenta algún error, indicara al vigilante “Sistema de Estudiante temporalmente inactivo”.
Requisitos especiales:	Como se mostrará el mensaje para permitir o denegar el acceso.
Frecuencia:	Alta, se van a realizarán un número elevado de consultas por día.
Temas Abiertos:	<ul style="list-style-type: none"> <li>› Manera en que se mostrara el mensaje.</li> <li>› Si el mensaje, desaparecerá después de un tiempo.</li> <li>› Mientras se encuentre el mensaje activo, no se podrá leer otro QR.</li> </ul>

## 2.1.4 SAR -&gt; Leer

	
Nombre de caso de Uso:	SAR -> Leer
Actor principal:	Escáner.
Precondiciones:	<ul style="list-style-type: none"> <li>› QR legible.</li> <li>› Extraer la dirección web guardada en el QR de manera correcta.</li> </ul>
Post condiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› Que las entradas, cuenten con un lector QR portátil.</li> </ul>
Escenario Principal:	Acceso 1.
Excepciones o Flujos Alternativos:	En caso de un error al leer el escáner, se deberá intentar de nuevo.
Requisitos especiales:	Conexión estable con el escáner QR.
Frecuencia:	Alto, van a leerse varios QR al día, en especial en los principales horarios de entrada.
Temas Abiertos:	<ul style="list-style-type: none"> <li>› Si el escáner utilizado, requiere un driver o API específico para su funcionamiento, o funciona al conectarlo a una computadora.</li> </ul>


## 2.1.5 Vigilante -&gt; Imprimir Acceso Temporal

	
Nombre de caso de Uso:	Vigilante -> Imprimir Acceso Temporal
Actor principal:	Vigilante
Precondiciones:	<ol style="list-style-type: none"> <li>1. Haber registrado al visitante.</li> <li>2. El registro, debe tener un identificador único.</li> <li>3. Que dicho registro, tenga un código QR asignado.</li> <li>4. El registro, se encuentre guardado en la DB.</li> </ol> <ul style="list-style-type: none"> <li>› Acceso a la DB.</li> <li>› Vista única, para la opción de impresión. Solo se imprimirán los que no sobrepasen las 4 horas de haberse registrado.</li> </ul>
Post condiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› Registro guardado en la DB.</li> <li>› QR asignado al registro.</li> <li>› Contar con una impresora.</li> </ul>




	<ul style="list-style-type: none"> <li>› Contar con un escáner, para la identificación única del usuario.</li> </ul>
Escenario Principal:	Vista única para impresión.
Excepciones o Flujos Alternativos:	<ul style="list-style-type: none"> <li>› Posible error con la impresora. En dicha condición, el personal de sistema o mantenimiento intervendrá.</li> </ul>
Requisitos especiales:	<ul style="list-style-type: none"> <li>› SI el QR ya venció, <b>no permitir la impresión</b>. Dentro de la UI, el registro ya no tendrá opción para imprimir.</li> <li>› En la DB, almacenara cuanto tiempo dura el código QR.</li> <li>› Después de las 7:00 p.m., la opción se <b>deshabilitará</b>, y se <b>prohibirá</b> cualquier acceso a visitantes.</li> </ul>
Frecuencia:	Medio-Bajo. Varía en cuestión de la actividad del plantel.
Temas Abiertos:	Se en la impresión, se imprimirá todos los datos del registro, los cuales se imprimirá en una hoja completa (tamaño carta).

### 2.1.6 Vigilante -> Registrar

	
Nombre de caso de Uso:	Vigilante -> Registrar
Actor principal:	Vigilante
Precondiciones:	<ul style="list-style-type: none"> <li>› Llenar todos los campos obligatorios.</li> <li>› Tener una cita previamente agendada.</li> <li>› Si no tiene una cita, dentro del apartado motivo, se especificará el motivo de la visita.</li> <li>› El visitante, obligatoriamente debe contar con una identificación oficial (INE).</li> </ul>
Post condiciones o Garantías de Éxito:	Los campos siguientes se consideran obligatorios: <ul style="list-style-type: none"> <li>› Nombre</li> <li>› Motivo</li> <li>› Identificación oficial (puede haber excepciones).</li> </ul>
Escenario Principal:	Acceso 1
Excepciones o Flujos Alternativos:	Si un usuario de la comunidad, olvido su credencial, este proporcionará su matrícula, se comprobará si existe y pertenece al usuario. De ser correcto, se le

	permitirá el acceso, y se guardará en el registro de visitantes.
Requisitos especiales:	<ul style="list-style-type: none"> <li>› Contar con una identificación oficial (INE), por parte del visitante, pueden existir casos especiales, en la cuales no sea requerida.</li> <li>› EL sistema, debe tener habilitada la opción para registrar. La opción se habilitará a las 6:59 a.m.</li> <li>› Después de las 7:00 p.m., la opción se <b>deshabilitará</b>, y se <b>prohibirá</b> cualquier acceso a visitantes.</li> </ul>
Frecuencia:	Medio-Bajo. Varía en cuestión de la actividad del plantel.
Temas Abiertos:	Si el usuario pertenece a la comunidad, y este olvido su credencial, que información se llenaría en "Motivo".

### 2.1.7 Visitante -> Validación Acceso Temporal

	
Nombre de caso de Uso:	Visitante -> Validación Acceso Temporal
Actor principal:	Visitante
Precondiciones:	<ul style="list-style-type: none"> <li>› El visitante debe estar registrado, y su QR debe encontrarse activo.</li> <li>› El visitante debe estar registrado y disponible durante el período de tiempo designado.</li> <li>› El sistema valida, si el QR está activo.</li> </ul>
Post condiciones o Garantías de Éxito:	<ul style="list-style-type: none"> <li>› El vigilante obtiene acceso al registro del visitante.</li> <li>› El sistema valida que el acceso esté habilitado durante la fecha y hora programadas.</li> <li>› El acceso del visitante, se encuentra registrado en la DB.</li> <li>› El sistema verifica que el acceso solicitado esté habilitado en ese momento (puede ser un evento, una reunión, acceso a un edificio, uso de un servicio, etc.).</li> </ul>
Escenario Principal:	Acceso 1
Excepciones o Flujos Alternativos:	<ul style="list-style-type: none"> <li>› <b>E1:</b> La identificación del visitante (código QR) no es válida.</li> </ul>

	<ul style="list-style-type: none"><li>• El sistema muestra un mensaje de error y solicita un nuevo escaneo o ingreso de datos.</li></ul> <ul style="list-style-type: none"><li>› <b>E2:</b> El acceso solicitado no está habilitado en la fecha o la hora actuales.<ul style="list-style-type: none"><li>• El sistema informa al visitante que el acceso no es posible debido a que no está disponible.</li></ul></li><li>› <b>E3:</b> El visitante no está registrado.<ul style="list-style-type: none"><li>• El sistema niega el acceso e informa al vigilante sobre el error de registro.</li></ul></li><li>› <b>E4:</b> El visitante intenta acceder fuera del horario permitido.<ul style="list-style-type: none"><li>• El sistema informa que el acceso está restringido fuera del horario autorizado.</li></ul></li></ul>
Requisitos especiales:	› Acceso a una red para validar el estado de acceso (en tiempo real si es necesario).
Frecuencia:	Medio-Bajo. Varía en cuestión de la actividad del plantel.
Temas Abiertos:	<ul style="list-style-type: none"><li>› Consideraciones de seguridad adicionales para garantizar que no se usen medios de identificación fraudulentos.</li><li>› Posibles problemas de validación por falta de conexión o fallas en el sistema.</li></ul>

## 2.2. Diagrama de Contexto

Diagrama de contexto, para el Sistema de Acceso Rápido (SAR), sistema el cual se va a desarrollar.



Diagrama de contexto general

Guadalupe Yamileth

1 hora

Diagrama 2

## 3. Punto de vista de la composición

### 3.1. Diagrama de paquetes

Mediante el diagrama de paquetes UML, se detallará el funcionamiento. El diagrama nos permitirá:

- Funcionamiento del sistema.
- Interfaces gráficas.
- Modelos y controladores.

Modelaremos la base de datos (DB) como un sistema aparte, esta representa las conexiones que realizara el sistema. Como el proyecto no tiene acceso completo a la información almacenada en la (DB), se tomó la decisión de representarlo cómo un sistema aparte.

El diagrama de paquetes seguirá la arquitectura Modelo-Vista-Controlador.

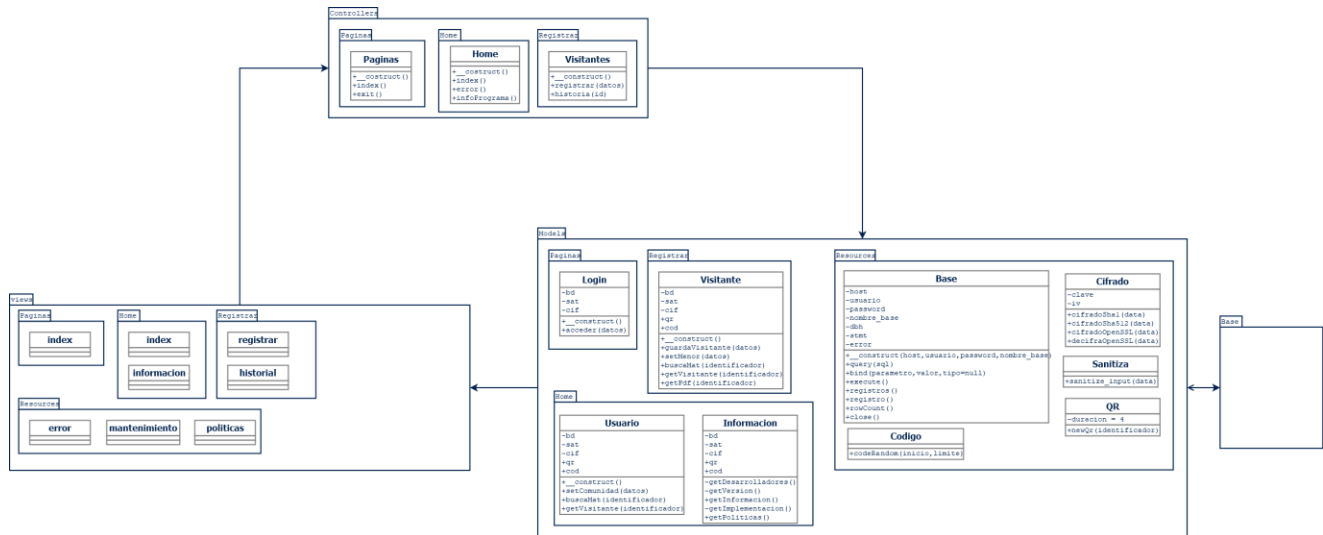


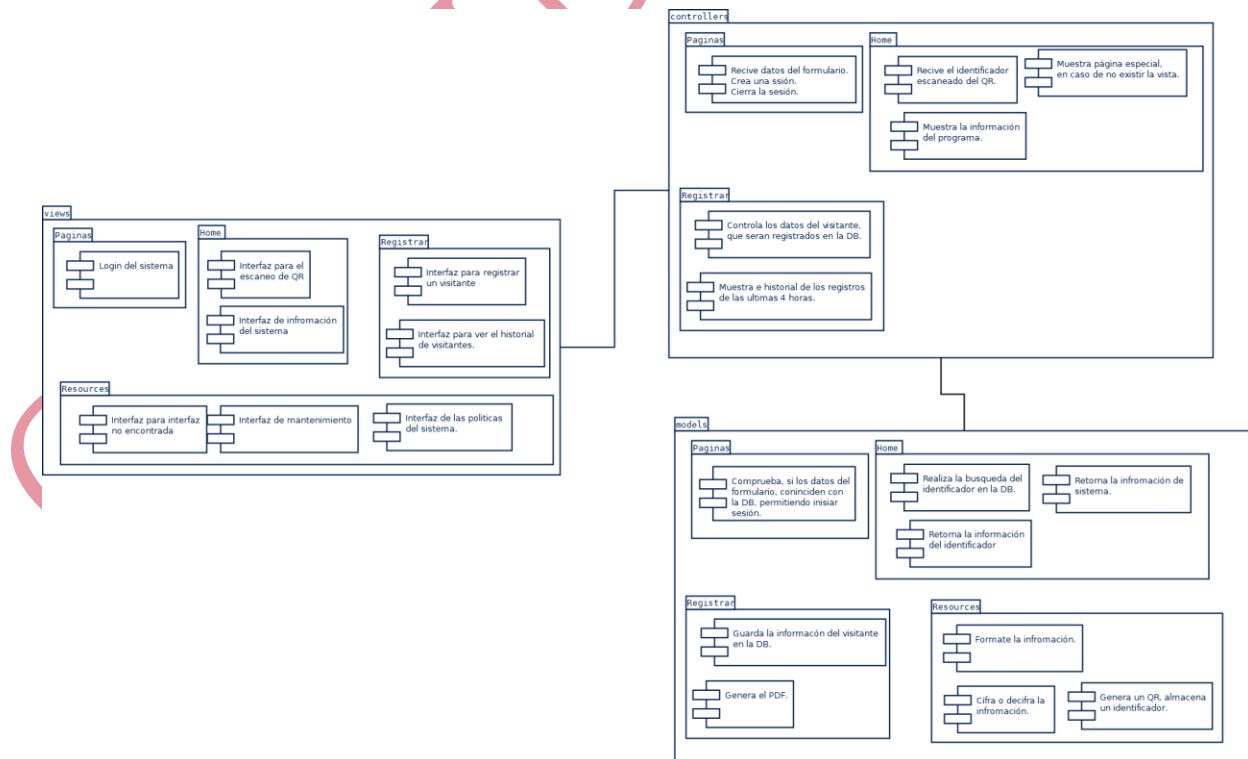
Diagrama general de paquetes

Manzur Rodríguez  
Diagrama 3

1.5 horas

### 3.2. Diagrama de componentes

El paquete 'controller', hace uso de 1 o varios modelos. En el diagrama, mencionamos todos los componentes que existen en cada paquete, los 3 paquetes, corresponden al MVC.



### 3.3. IDEFO

Se describe el proceso de entrada de estudiantes, personal administrativo y trabajadores a la universidad pública. La entrada está regulada por los vigilantes de seguridad, quienes verificarán las credenciales con la ayuda de un nuevo programa en desarrollo.

#### 3.3.1 Sistema actual

El sistema que sigue actualmente la universidad, para permitir el acceso a estudiantes, personal administrativo, docentes, trabajadores al plantel, se basa en el uso de la credencial, que previamente se les proporcione, como identificación oficial por parte de la UACM.



Diagrama IDEF0

#### 3.3.2 Nuevo sistema

A pesar que el sistema actual es funcional, presenta cuestiones importantes, en el tema de seguridad, esto, a que no se comprueba la información de dichas credenciales, por parte de los vigilantes. El sistema (SAR), busca mejorar la seguridad, en el acceso al plantel, y al mismo tiempo, ayudar a que los vigilantes, tengan mejores herramientas, para dejar entrar a un usuario al plantel.

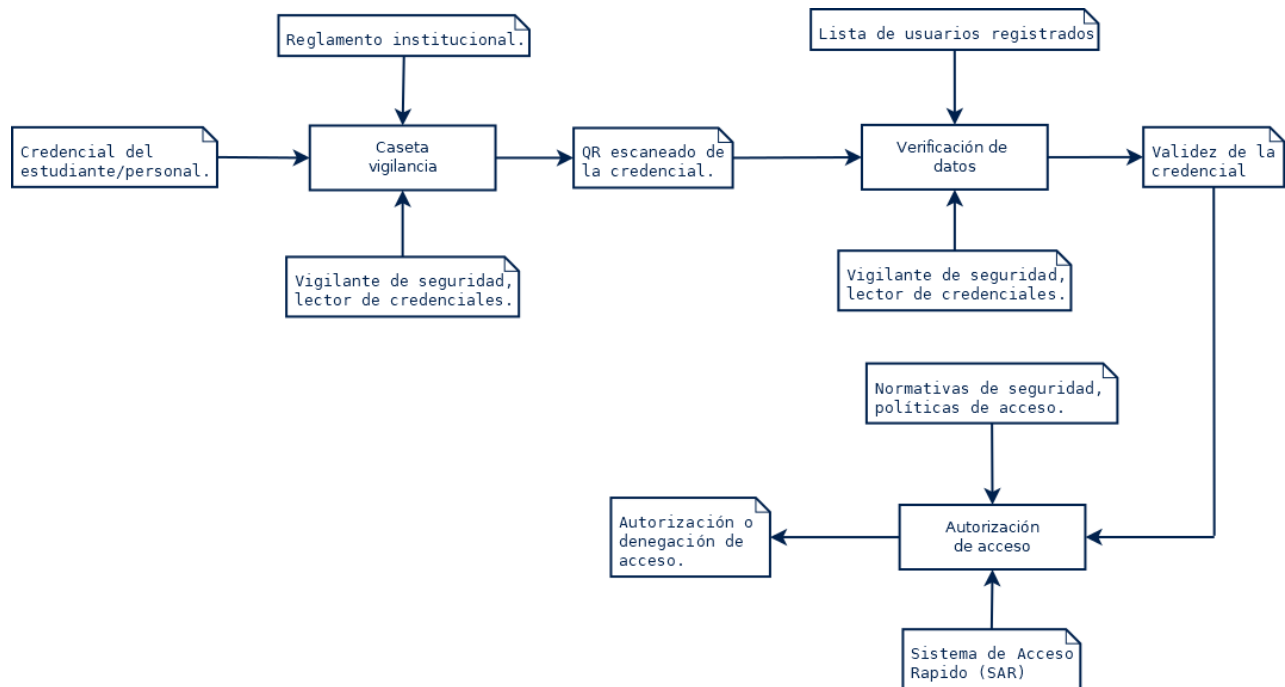


Diagrama IDEF0 SAR

Manzur Rodríguez

2 horas

Diagrama 6

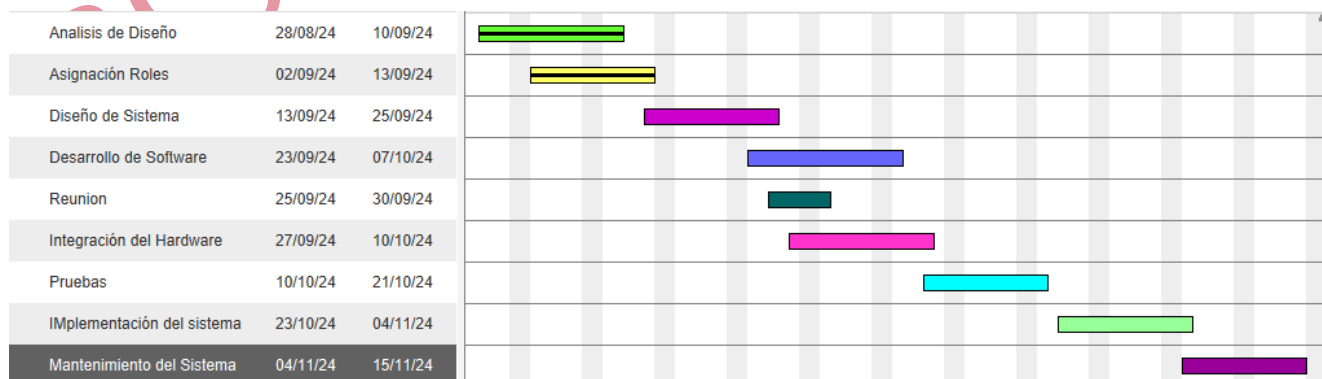
### 3.4. Diagrama GANTT

Para realizar la planificación, supervisar su evolución, retroalimentaciones o actualizaciones por parte del cliente o gerente del proyecto, se utilizará la herramienta ‘GanttProjet’.

Utilizaremos la versión gratuita de la herramienta, por lo tanto, las actualizaciones o correcciones, se realizarán en las reuniones. Con el objetivo, que todo el equipo de desarrollo tenga conocimiento de alguna actualización o modificación de los requisitos.

Esta herramienta nos permitirá estimar el costo, el personal que será asignado y el cronograma para el desarrollo.

**NOTA:** Todo el equipo de desarrollo trabajará en días laborales (lunes – viernes). Por lo tanto, el equipo de desarrollo no trabajará los fines de semana.



Las etapas de desarrollo se asignará el trabajo de manera equitativa, el tiempo estimado de desarrollo, que tiene el equipo:

### 3.4.1 Estimación de Tiempo

- › Análisis de Requerimientos
  - Duración Estimada: 2 semanas.
- › Diseño del Sistema
  - Duración Estimada: 3 semanas.
- › Desarrollo
  - Duración Estimada: 8 semanas.
- › Pruebas
  - Duración Estimada: 4 semanas.
- › Implementación
  - Duración Estimada: 2 semanas.
- › Soporte Post-Implementación
  - Duración Estimada: 2 semanas.

Utilizando la aplicación antes mencionada, establecimos los roles del equipo de desarrollo, las fechas límite para la codificación. Con esto, podemos determinar de manera visual si un desarrollador tiene sobrecarga de trabajo.

## 3.5. Entidad de diseño

### 3.5.1 Patrón

El proyecto será diseñado utilizando el patrón de desarrollo Modelo – Vista – Controlador (MVC). Se escogió dicho patrón, con el objetivo de facilitar las actualizaciones, o mejoras posteriores del sistema.

### 3.5.2 Framework

Se utilizará el Framework ‘Laravel’, en su versión 11.0

## 3.6. Diagrama HIPO



Usando el diagrama HIPO, modelaremos las acciones a realizar por parte de (SAR). Derivado del análisis estructurado y la técnica de diseño (SADT). Este método nos ayudara a realizar el análisis del sistema y a promover una buena comunicación entre el análisis previo del sistema, por parte de los desarrolladores y el cliente.

Utilizaos los rectángulos y líneas para representar procesos, funciones, trabajos o tareas, así como las conexiones existentes entre las funciones y el entorno externo.

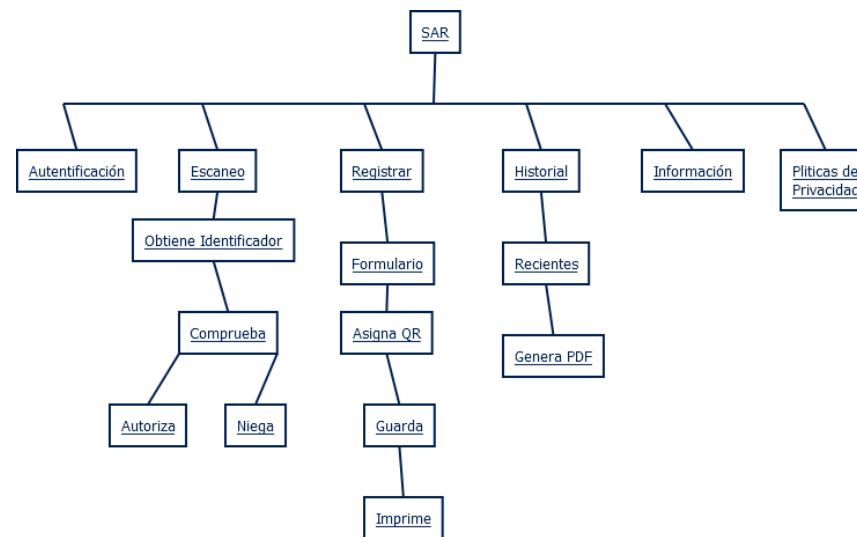


Diagrama HIPO

Manzur Rodriguez

1.5 Horas

Diagrama 8

## 4. Punto de vista lógico

### 4.1. Propósito

El propósito del punto de vista lógico en este diagrama es identificar los objetos dentro del sistema de control de acceso mediante QR y sus relaciones estáticas. Se diseñan clases y objetos que interactúan para representar la autenticación de usuarios y visitantes mediante códigos QR.

### 4.2. Problemas de diseño

El diagrama de objeto, los principales problemas de diseño abordados incluyen:

- › Identificación de entidades claves: Se define objetos como **Usuario**, **Visitante**, **Login**, **Sistema**, **GeneradorQR**, **QRcode**, **Lector** y **Escaneable**.
- › **Reutilización de abstracciones**: Se presentan objetos que pueden ser utilizados en distintas instancias del sistema, como **Login** para diferentes usuarios o **QRCode** generado para cada visitante.
- › **Interacción entre los elementos**: Se requiere definir cómo se comunican los objetos entre sí, como la relación entre el **GeneradorQR**, el **QRCode** y el **Lector**.

## 4.3. Elementos de diseño

### 4.3.1 Entidades de diseño

- › **Clases y objetos**

Se identifican clases implícitas en el diagrama como **Usuario**, **Visitante**, **Login**, **Sistema**, **GeneradorQR**, **QRCode**, **Lector** y **Escaneable**.

- › **Atributos**

Algunos atributos incluyen usuario y contraseña en **Login**, código en **QRCode** y motivo en **Visitante**.

- › **Métodos**

Aunque no se especifican explícitamente, el **GeneradorQR** tiene un método para generar códigos QR y **Lector** uno para escanearlos.

### 4.3.2 Relaciones de diseño

- › **Asociación:**

Tiene una relación entre **Usuario** y **Sistema**, indicando que el sistema maneja datos del usuario.

- › **Generalización**

**Escaneable** es una clase abstracta que puede representar cualquier objeto escaneable, como un código QR.

- › **Dependencia**

**Lector** depende de **Escaneable**, lo que indica que **Lector** no tiene sentido sin un objeto que pueda escanear.

### 4.3.3 Atributos de diseño

- › **Nombre y rol**

Cada objeto tiene un identificador y atributos relevantes.

- › **Cardinalidad**

Se infiere que puede haber múltiples **Usuarios**, **Visitantes** y **QR Codes**, pero el **Sistema** actúa como un contenedor único.

### 4.3.4 Restricciones de diseño

- › **Multiplicidad**

Un **Usuario** visitante puede autenticarse varias veces, y no generar múltiples **QRCode**.

## › Navegabilidad

La relación entre Lector y QRCode sugiere una dirección clara: Lector escanea QRCode, pero no al revés.

### 4.4. Ejemplos de idiomas

#### › Diagrama de objetos UML

Representa instancias específicas de las clases, como Usuario con matrícula 20231025 y Visitante con motivo = 'Reunión'.

#### › Diagramas de clases UML

Podrían utilizarse para representar la estructura general, definiendo Usuario y Visitante como subclases de una clase Persona, por ejemplo.

### 4.5. Diagrama de clases UML

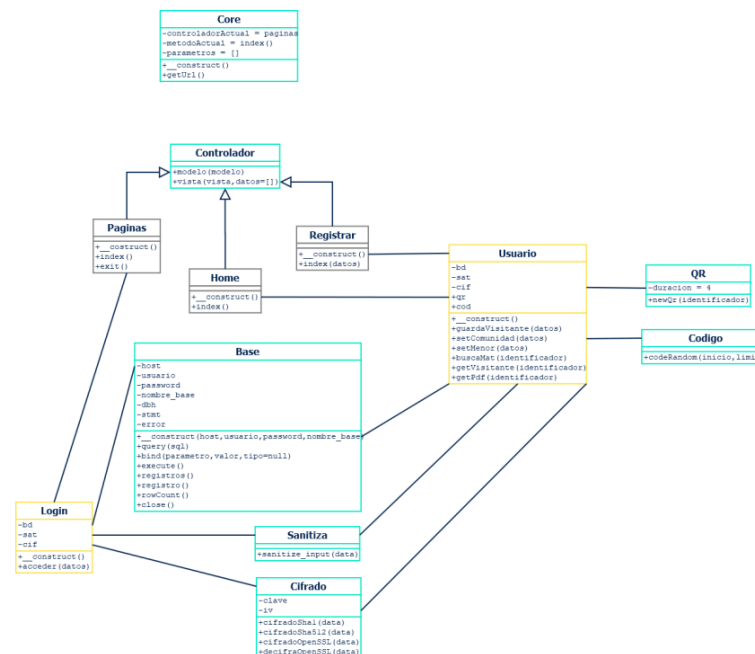


Diagrama clase

Kevin Manzur

1.5 Horas

Diagrama 9

### 4.6. Diagrama de objeto UML

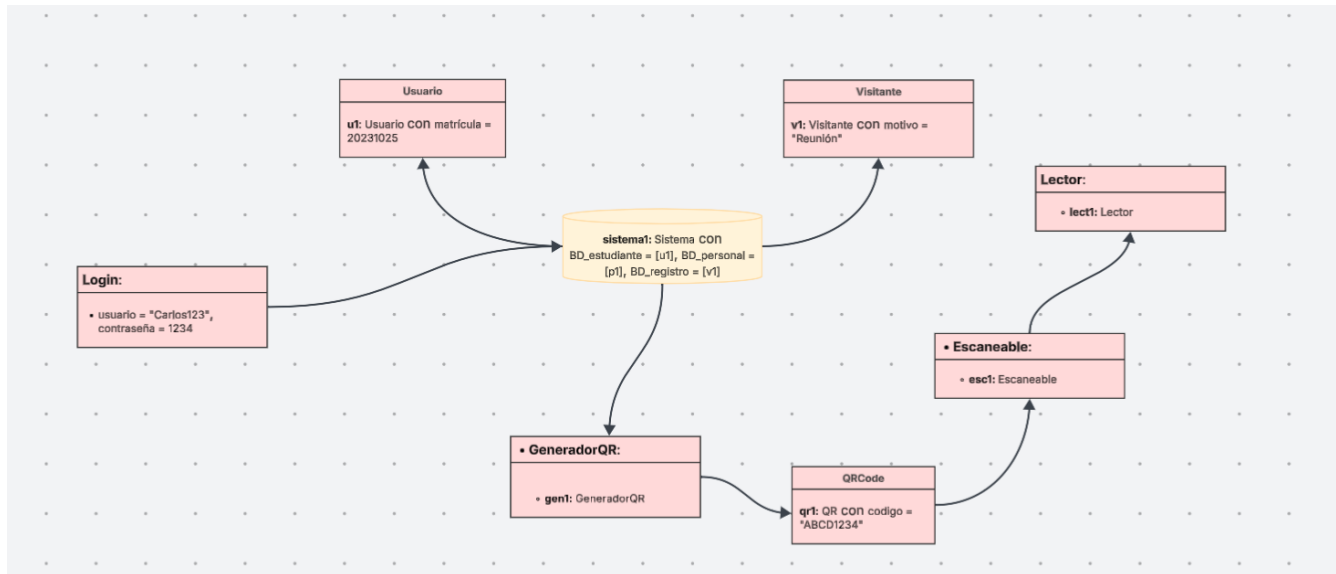


Diagrama objeto

Cristela Cruz Ovando  
Diagrama 10

3.5 Horas

## 5. Punto de vista de la dependencia.

La vista de dependencias tiene como objetivo describir las relaciones entre los diferentes elementos del sistema, mostrando cómo interactúan y dependen unos de otros. En este apartado, se detallan las conexiones entre los principales componentes del sistema de acceso a la UACM mediante QR, incluyendo módulos de autenticación, gestión de usuarios y validación de accesos.

Esta vista es esencial para comprender el nivel de acoplamiento entre los elementos del software, identificar posibles mejoras en la arquitectura y garantizar que el diseño del sistema sea flexible y mantenible. Para ello, se presentan los diagramas de componentes y paquetes, los cuales ilustran las interacciones y agrupaciones lógicas dentro del sistema.

### 5.1. Problemas de diseño

En el diseño del sistema de acceso a la UACM mediante QR, se han identificado los siguientes problemas de diseño:

- › **Acoplamiento entre componentes:** La interdependencia entre controladores y la base de datos puede generar problemas de mantenimiento y escalabilidad.
- › **Gestión de autenticación:** La seguridad en la comunicación entre el componente de Login y la base de datos requiere especial atención para evitar vulnerabilidades.
- › **Manejo de sesiones:** La conexión entre los componentes Home y QR debe garantizar que los accesos sean seguros y eficientes.

#### 5.1.1 Elementos de diseño

Los principales elementos de diseño incluyen:

- › **Componentes del sistema:** Representados en el diagrama de componentes, incluyen los módulos principales como Login, QR, Usuario, Visitante y los controladores correspondientes.
- › **Relaciones entre componentes:** Definen cómo interactúan los diferentes elementos del sistema, incluyendo la base de datos como punto central de la gestión de información.
- › **Estructura modular:** Utilizando Laravel, los controladores se encargan de la lógica de negocio y se comunican con la base de datos mediante modelos.

### Diagrama de Componentes

El diagrama de componentes muestra la estructura del sistema, detallando los módulos principales y su interacción con los controladores y la base de datos.

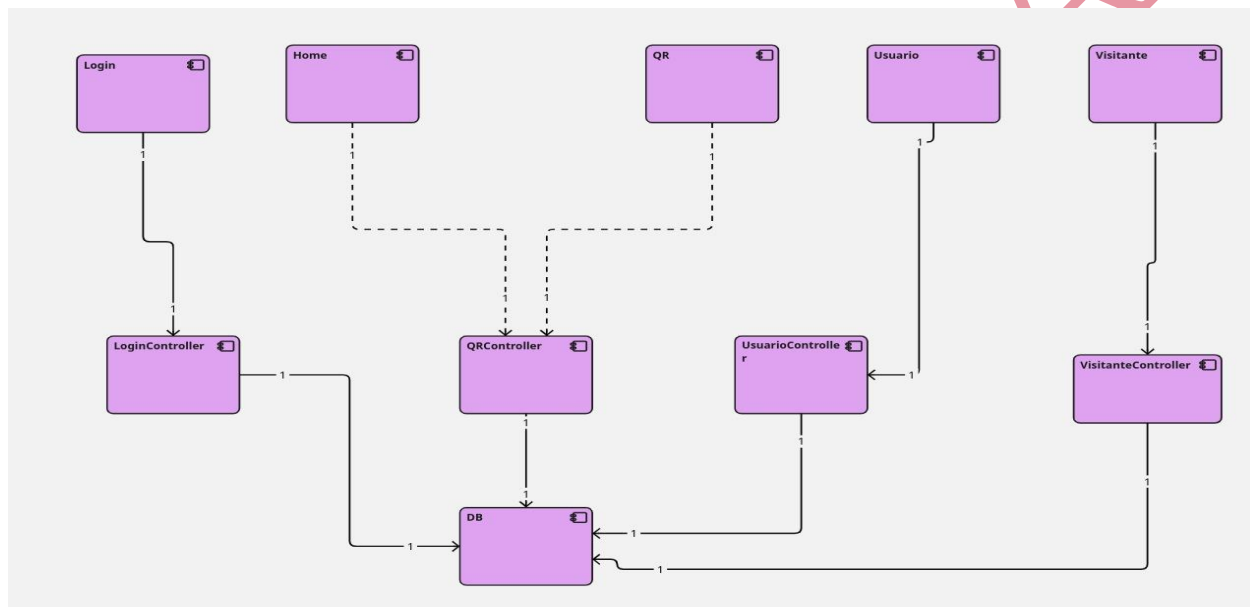


Diagrama de componentes

Guadalupe Yamileth

1.5 Horas

Diagrama 11

### 5.1.2 Atributos de dependencias

Las dependencias dentro del sistema están determinadas por:

- › Tipo de dependencia: Comunicación entre controladores y base de datos, dependencias entre módulos de usuario y autenticación.
- › Direccionalidad: Se identifican relaciones unidireccionales y bidireccionales según la interacción de los componentes.
- › Grado de acoplamiento: Se busca minimizar dependencias innecesarias para mejorar la escalabilidad del sistema.

## 5.2. Ejemplos de Idiomas

El sistema está desarrollado en PHP, con Laravel como framework. Se utilizan estándares de desarrollo web, incluyendo HTML, CSS y JavaScript para la interfaz de usuario.

## Diagrama de Paquetes

El diagrama de paquetes organiza los componentes en módulos lógicos, reflejando la estructura del sistema y facilitando su mantenimiento.

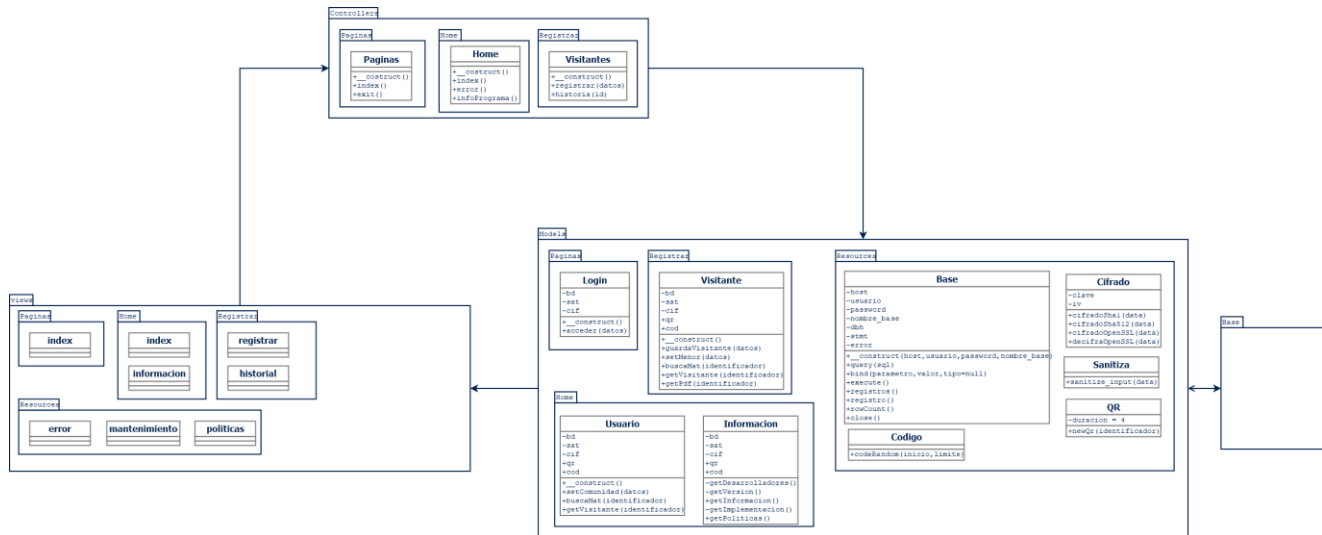


Diagrama general de paquetes

Manzur Rodriguez  
Diagrama 12

3 horas

## 6. Punto de vista de uso de patrones

El patrón **Modelo-Vista-Controlador (MVC)** permite organizar el desarrollo de software separando la lógica de datos, la interfaz y el control del sistema. En el proyecto actual de **SAR**, este patrón mejora la claridad y el mantenimiento del código.

**Laravel**, un framework basado en MVC, facilita este tipo de desarrollo al ofrecer herramientas para enrutamiento, autenticación y manejo de base de datos. Usar Laravel con MVC permite construir un sistema eficiente, seguro y escalable para controlar el acceso de estudiantes, personal y visitantes mediante escaneo de códigos QR.

Laravel **separa y organiza** claramente:

- **Modelo** → Datos.
- **Vista** → Interfaz.
- **Controlador** → Lógica.

Todo esto **automatiza el patrón MVC** para que solo nos enfoquemos en la lógica del proyecto.

### 6.1.1 Problemas de diseño

Al desarrollar el sistema **SAR**, pueden surgir problemas clave como la necesidad de **reutilizar ideas de diseño**, aplicar **estilos arquitectónicos** adecuados y aprovechar **plantillas de frameworks**.

El patrón **MVC**, junto con Laravel, ayuda a organizar el código, facilita el mantenimiento y mejora la eficiencia del desarrollo al resolver estos problemas de forma estructurada.

### 6.1.2 Elementos de diseño

En el sistema **SAR** usando MVC y Laravel:

- **Entidades de diseño:** incluyen clases, roles, conectores (como rutas), y el uso de Laravel como plantilla de marco.
- **Relaciones de diseño:** se reflejan en asociaciones entre clases y la colaboración entre vista, controlador y modelo.
- **Atributos de diseño:** como los nombres claros de clases y funciones.
- **Restricciones de diseño:** se aplican para que la vista no acceda directamente al modelo, respetando la estructura MVC.

### 6.1.3 Ejemplos de idiomas

Se utilizan **idiomas visuales** como los diagramas UML para representar la estructura del sistema.

- **Diagrama de estructura compuesta UML:** muestra cómo los componentes internos (modelo, vista, controlador) se relacionan dentro del sistema.

## 6.2. Diagrama de Estructura Compuesta UML

El diagrama de estructura compuesta muestra cómo interactúan los componentes internos dentro del patrón MVC.

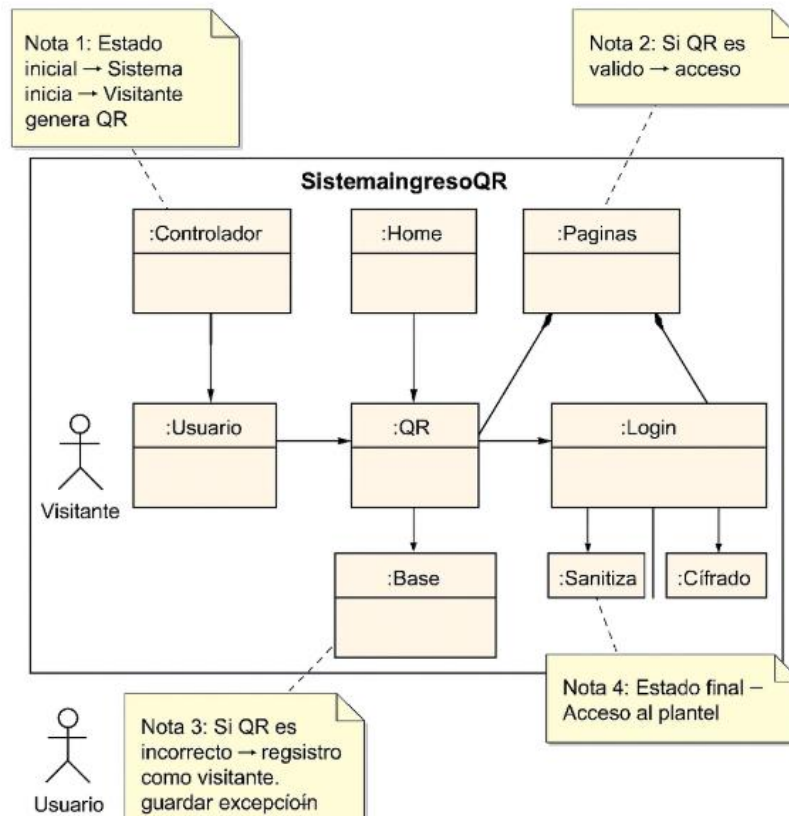


Diagrama de estructura compuesta UML

Cristela Cruz Ovando

3.5 Horas

Diagrama 13

## 7. Definiciones, acrónimos y abreviaturas

- › **SAR:** “Sistema de Acceso Rápido” (SAR).
- › **UACM:** Universidad Autónoma de la Ciudad de México.
- › **Comunidad:** Estudiantes y trabajadores.
- › **Trabajadores:** Personal docente e investigador, Personal de administración y servicios y Personal de vigilancia.
- › **Campus:** Área de instalaciones universitarias donde se realizan actividades académicas y administrativas.
- › **Servidor:** Sistema informático que proporciona recursos y servicios a otros ordenadores a través de una red.
- › **Base de Datos:** Conjunto organizado de datos almacenados electrónicamente, permitiendo su gestión y actualización.
- › **Normativas:** Reglas y directrices establecidas por una autoridad para regular comportamientos y acciones.
- › **Políticas:** Normas que regulan las actividades y comportamiento dentro de la institución.



- › **UI (User Interface):** UI significa Interfaz de Usuario. Se refiere a la parte del software con la que los usuarios interactúan directamente. El diseño de UI se enfoca en la disposición visual y la presentación de los elementos en la pantalla.
- › **UX (Experiencia de Usuario):** UX Se refiere a la experiencia general del usuario al interactuar con el software. El diseño de UX abarca aspectos más amplios que solo la apariencia y se centra en cómo se siente el usuario durante el uso del producto.
- › **QA (Aseguramiento de la Calidad):** Es un proceso integral que se enfoca en asegurar que el software cumpla con los estándares de calidad y que funcione correctamente según los requisitos definidos.
- › **Formador:** Es un profesional encargado de capacitar a los usuarios, desarrolladores, y otros miembros del equipo sobre el uso de software, herramientas o metodologías específicas.
- › **Visitante:** Usuario final, el cual no pertenece de ninguna manera al plantel educativo.
- › **DB:** Base de datos de la UACM.
- › **Usuario:** Persona que desea acceder a la institución educativa.

## 8. Bibliografía

- A.U.S. Gustavo Torossi. Diseño de Sistemas. El proceso unificado de desarrollo de Software.
- Cervantes, Velasco, Castro; Arquitectura de Software. Conceptos y Ciclo de Desarrollo; Cengage Learning, 2016.