

# UACM

Universidad Autónoma  
de la Ciudad de México

**NADA HUMANO ME ES AJENO**

## **Diagramas**

ANALISIS Y MODELAMIENTO DE SOFTWARE

### **Entrada estudiantil mediante QR**

Valadez Carmona Guadalupe Yamileth

Rodríguez Cervantes Kevin Manzur

Cruz Ovando Cristela Adelaida

Rodríguez Sánchez Diana Fabiola

Romero Cervantes Fátima Daniela

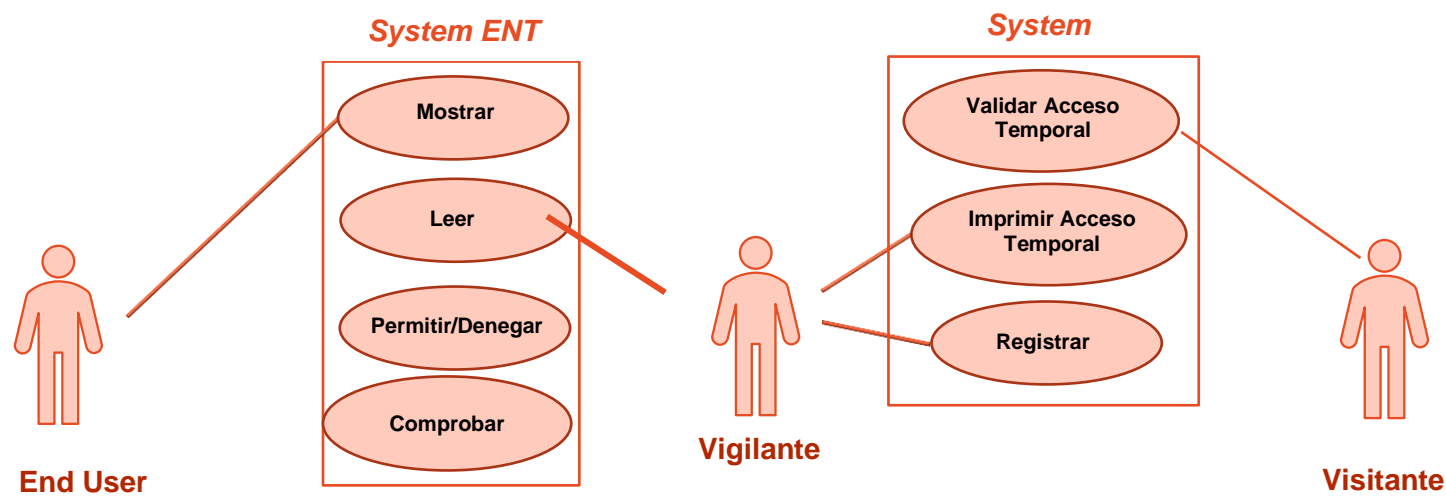
# HISTORIAL DE VERSIONES

Fecha	Versión	Descripción	Autores
14-10-2024	0.5	<ul style="list-style-type: none"><li>○ Versión preliminar de diagramas</li></ul>	Guadalupe Yamileth, Cristela Adelaida Diana Fabiola, Kevin Manzur, Fátima Daniela
15-10-2024	1.10	<ul style="list-style-type: none"><li>○ Caso de uso</li><li>○ Diagrama de clase</li><li>○ Diagrama de secuencia general.</li><li>○ Diagrama de secuencia por clase.</li></ul>	Guadalupe Yamileth, Cristela Adelaida Diana Fabiola, Kevin Manzur, Fátima Daniela
22-11-24	1.20	<ul style="list-style-type: none"><li>○ Actualización del diagrama de clases.</li><li>○ Actualización de los diagramas de secuencia.</li><li>○ Nueva herramienta implementada para los diagramas de secuencia.</li></ul>	Guadalupe Yamileth, Cristela Adelaida Diana Fabiola, Kevin Manzur, Fátima Daniela

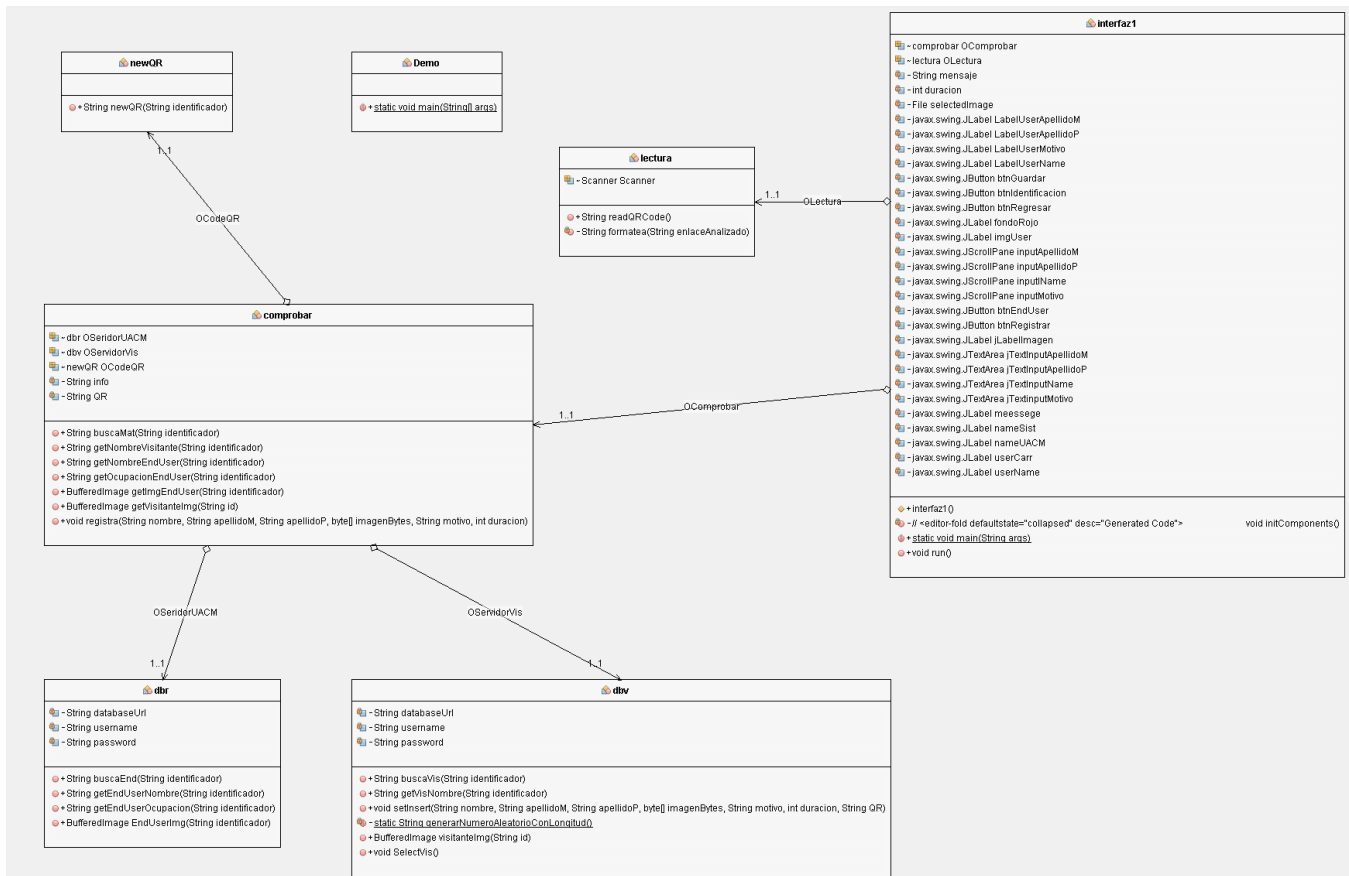
## Índice

Caso de Uso.....	1
Diagrama de clase.....	2
Diagrama de secuencia .....	3
Diagrama buscaMat.....	4
Diagrama dbr .....	5
Diagrama dbv .....	6
Diagrama buscaEnd .....	7
Diagrama lectura.....	8
Diagrama newQR .....	9
Diagrama registra .....	10
Diagrama buscaVis.....	11
Definiciones, acrónimos y abreviaturas.....	12
Bibliografía.....	13

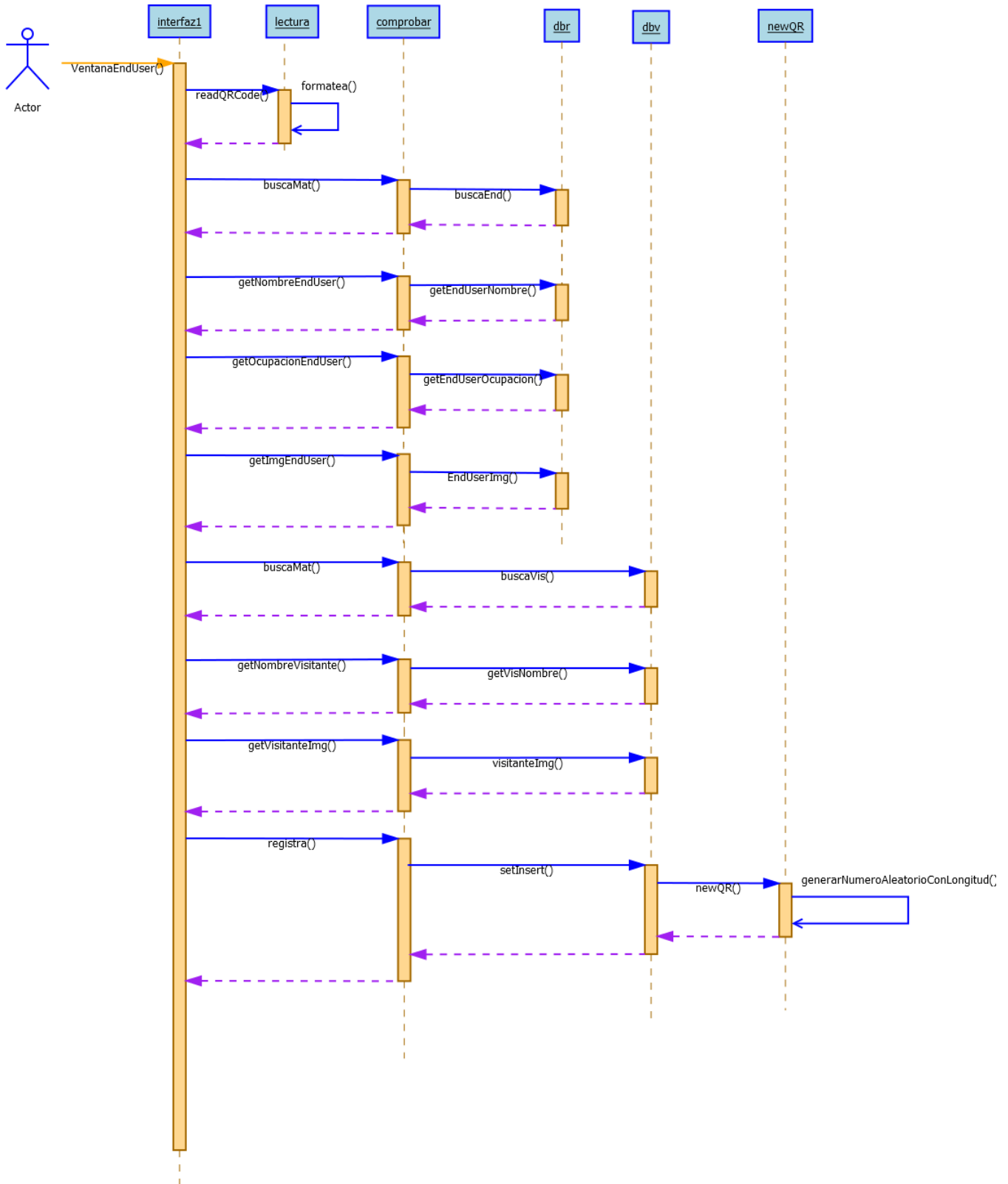
# Caso de Uso



# Diagrama de clase



# Diagrama de secuencia



## Diagrama buscaMat

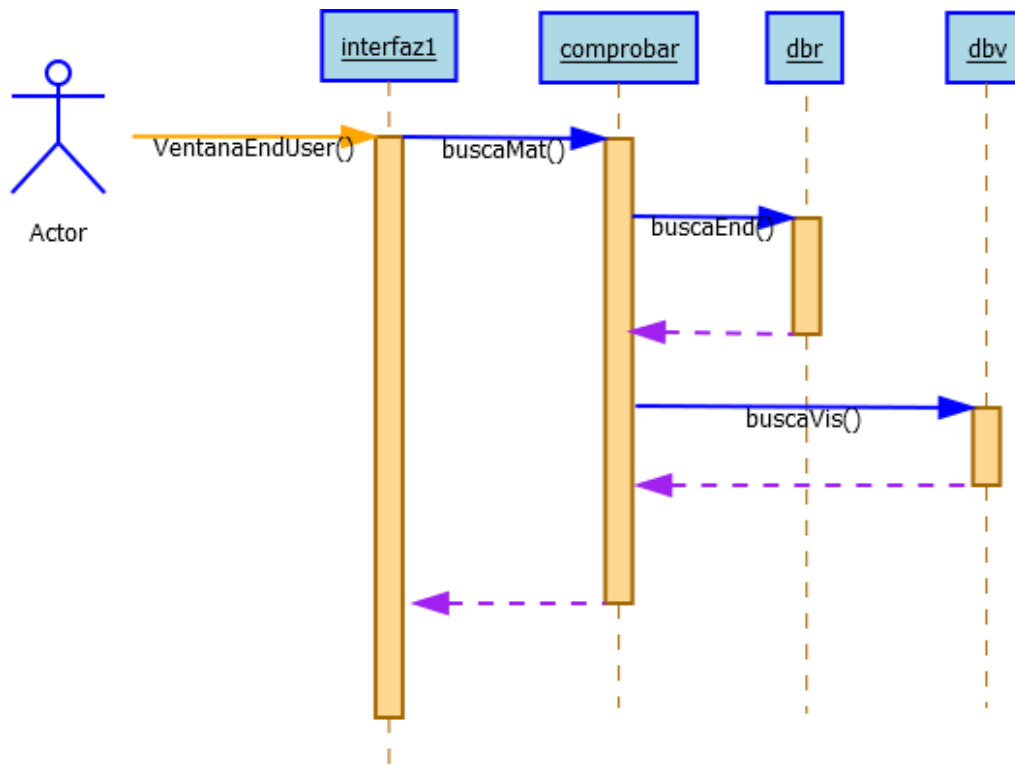


Diagrama	buscaMat
Realiza la búsqueda de un identificador, en la base de datos dbr (End User), y en la base de datos dbv (visitantes).	
Las dos DB ya se encuentran implementadas en los servidores de la UACM, por lo tanto, el personal de Sistemas nos proporcionara un usuario, con el propósito de realizar consultas, ingresar datos o actualizaciones.	
Método	Descripción
buscaMat()	Realiza la búsqueda de un identificador (is - matricula) en la base de datos
buscaEnd()	Busca un identificador (id), en la base de datos, la cual es exclusivamente para los End USer.
buscaVis()	Busca un identificador (id), en la base de datos, la cual es exclusivamente para los Visitantes.

## Diagrama dbr

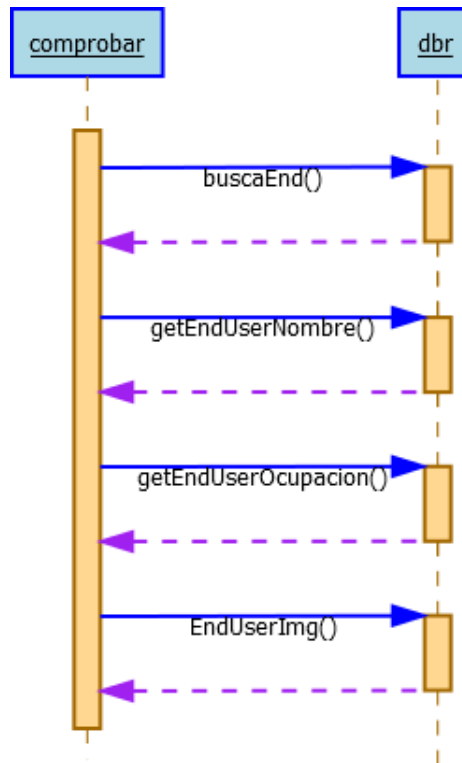


Diagrama	dbr
<p>Es la base de datos que almacena los datos principales de los estudiantes que pertenecen al plantel de Cuauhtémoc.</p> <p>Dicha base de datos ya existe y es la que el sistema estudiantil utiliza para verificar si un estudiante sigue activo o no, y en caso de que lleve mucho tiempo activo su matrícula es suspendida, por lo que nos enfocaremos en las matrículas activas que aparezcan en la base de datos de la universidad.</p>	
Método	Descripción
buscaEnd()	Realiza una búsqueda en la base de datos, comprueba si existe el identificador en la columna matrícula. De existir, retorna un string, informando que sí existe; en caso contrario, retorna un string indicando que no existe.
getEndUserNombre()	Retornamos el nombre completo del End User almacenado en la DB.
getEndUserOcupacion()	Retornamos la ocupación del End User.
EndUserImg()	Obtiene la fotografía almacenada en la DB, y la retorna para ser mostrada en la interfaz.



## Diagrama dbv

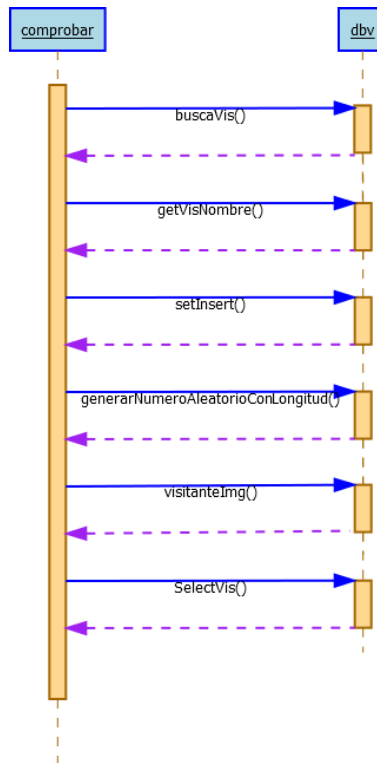


Diagrama	dbv
Es la base de datos que será utilizada para llevar un control de las personas ajenas al plantel que deseen entrar para poder disfrutar de alguna actividad que esté ofreciendo la universidad en el plantel de Cuauhtepic.	
Método	Descripción
buscaVis()	Realiza una búsqueda en la base de datos, comprueba si existe el identificador en la columna id. De existir, retorna un string, informando que sí existe; en caso contrario, retorna un string indicando que no existe.
getVisNombre()	Retorna el nombre completo almacenado en la base de datos, solo retorna el nombre que coincida con el identificador numérico pasado como parámetro.
getInsert()	Registra a un nuevo visitante, y guarda su información a la DB.
generarNumeroAleatorioConLongitud()	Genera un número aleatorio con una longitud aleatoria entre 5 y 15
visitanteImg()	Muestra la imagen de la identificación proporcionada en su registro.
SelectVis()	Retorna los registros, del mismo día en que se esté utilizando. No muestra registros de días anteriores.

## Diagrama buscaEnd

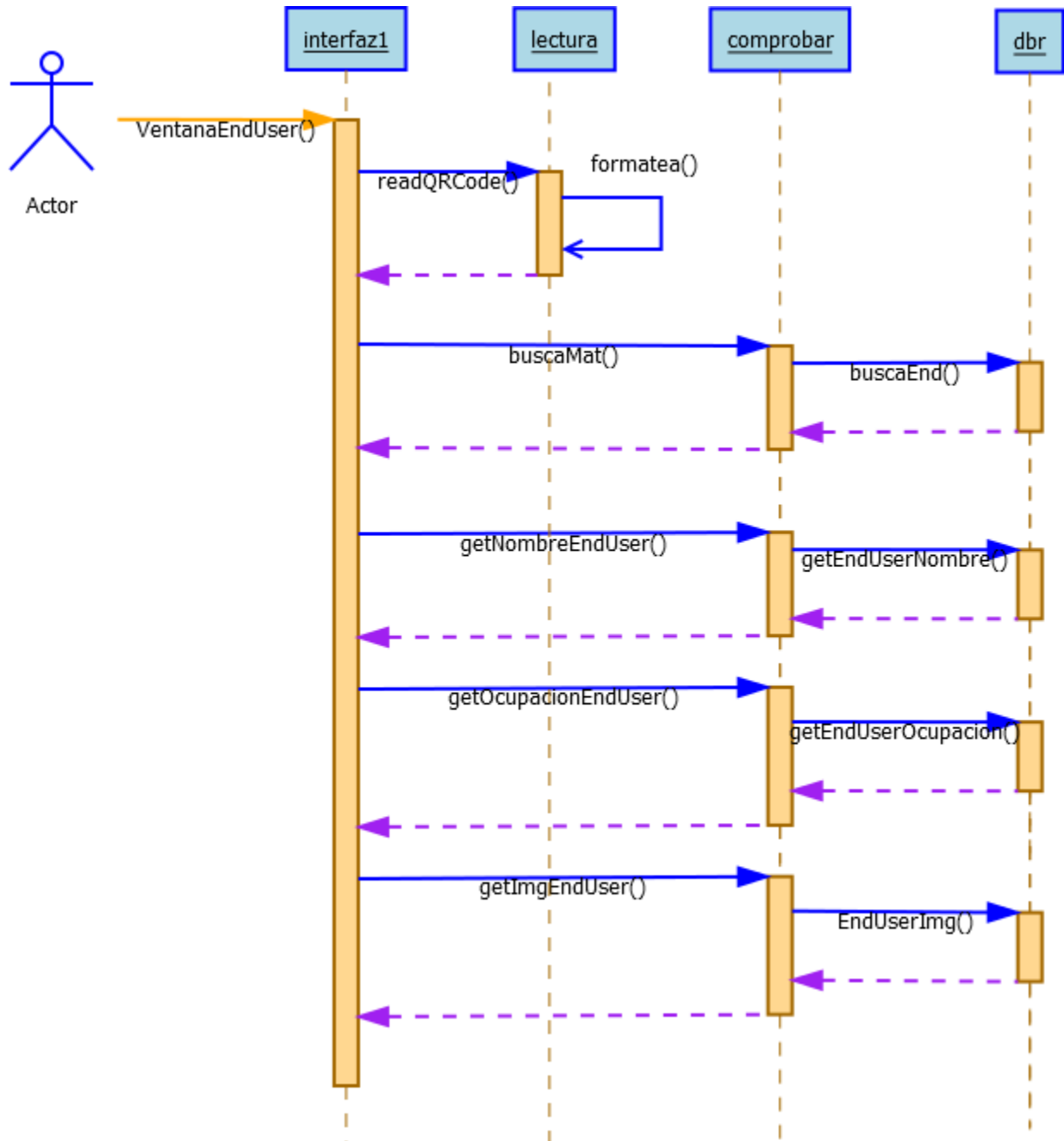


Diagrama	endEncontrado
Está función nos permitirá saber si el alumno se encuentra registrado en la base de datos de la universidad en el plantel de Cuauhtepac.	
Método	Descripción
VentanaEndUser()-> readQRCode()	Se conecta con el escáner, el cual lee un código QR. Se procesa el enlace y retorna el identificador.

<code>readQRCode()</code> -> <code>formatea()</code>	Formatea el enlace web obtenido, extrae el identificador almacenado dentro del enlace y retorna solo el identificador.
<code>VentanaEndUser()</code> -> <code>buscaMat()</code> -> <code>buscaEnd()</code>	Del identificador obtenido por el escáner, comprueba si este existe en alguna de las bases de datos. De existir, retorna un Sting, indicando en qué DB encontró el identificador.
<code>VentanaEndUser()</code> -> <code>getNombreEndUser()</code> -> <code>getEndUserNombre()</code>	Envía como parámetro el identificador, este se lo envía a las demás clases, el tipo de dato retornado es un Sting, el cual contiene el nombre completo de un End User.
<code>VentanaEndUser()</code> -> <code>buscaMat()</code> -> <code>getOcupacionEndUser()</code>	Envía por parámetro el identificador, este al final es envía a la DB, en el cual realiza la búsqueda para obtener la asignación.
<code>VentanaEndUser()</code> -> <code>getNombreEndUser()</code> -> <code>getImgEndUser()</code>	Envía por parámetro el identificador, este al final es envía a la DB, en la cual realiza la búsqueda para obtener la fotografía almacenada en la base de datos.

### Diagrama lectura

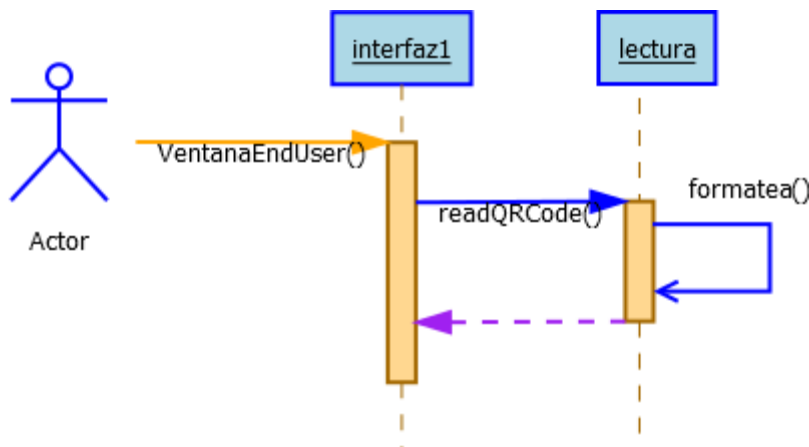


Diagrama	lectura
Nos permitirá leer los códigos QR que nos proporcionen tanto los alumnos como los visitantes que deseen ingresar al plantel.	
Método	Descripción
<code>VentanaEndUser()</code> -> <code>readQRCode()</code>	El escáner se encuentra activo, se va a escanear un código QR, el cual se obtendrá un enlace. Para aceptar el enlace, se deben de cumplir ciertas restricciones.
<code>readQRCode()</code> -> <code>formatea()</code>	Formatea el enlace web obtenido, extrae el identificador almacenado dentro del enlace y retorna solo el identificador.

VentanaEndUser()-> Formatea()	Antes de retornar la información, se formatea la información y se extrae el identificador, este es el que retorna.
-------------------------------	--

### Diagrama newQR

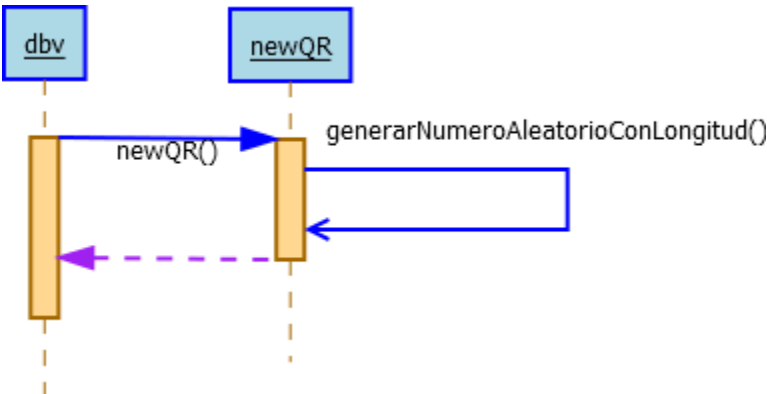


Diagrama	newQR
Nos va a permitir generar nuevos QR para que los visitantes que deseen ingresar a disfrutar de las actividades proporcionadas por el plantel puedan ingresar a las instalaciones.	
Método	Descripción
newQR()	Será generado después de que el visitante sea registrado con motivo de su entrada y solo entonces obtendrá su QR. Este almacena un enlace web, en la dirección web contiene el identificador asignado a su registro.
newQR() -> generarNumeroAleatorioConLongitud()	Genera un número aleatorio con una longitud aleatoria entre 5 y 15

## Diagrama registra

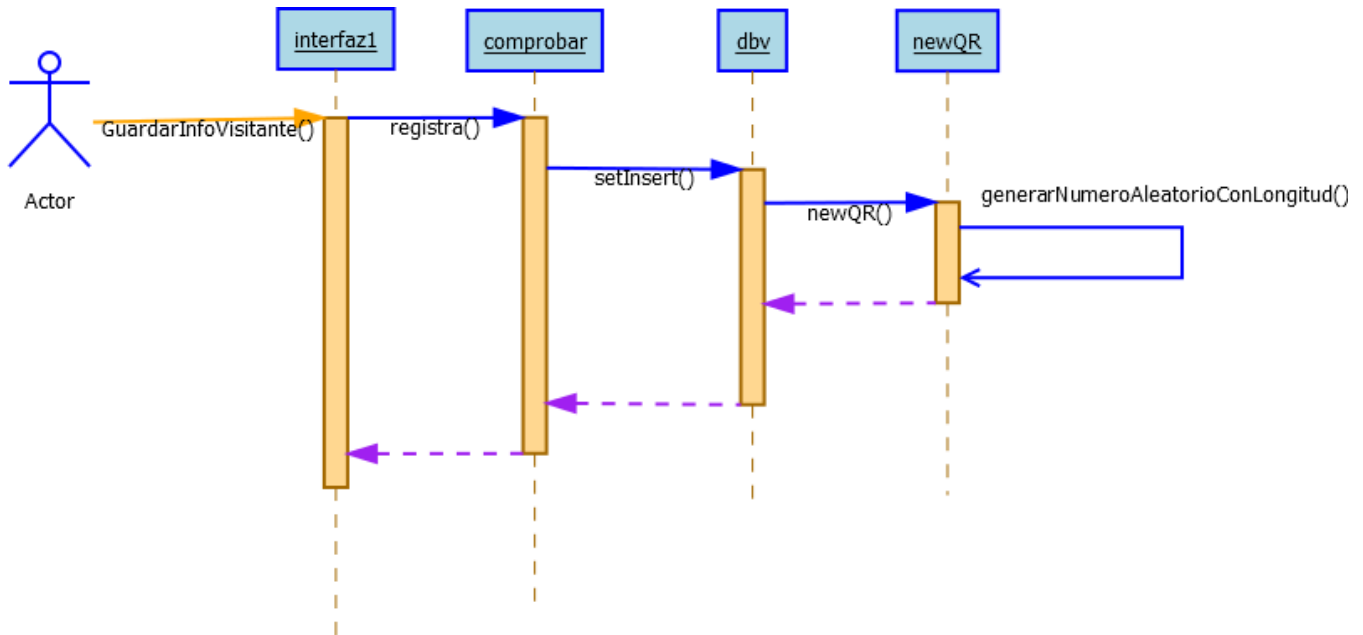


Diagrama	registra
Esta función estará habilitada solamente para el vigilante, que es el que va a registrar la información de las personas ajenas al plantel que deseen ingresar por un motivo en específico.	
Método	Descripción
GuardarInfoVisitante() -> registra()	El visitante será registrado después de proporcionar sus datos al vigilante.
GuardarInfoVisitante() -> newQR()	Genera un nuevo QR con el identificador asignado al registro.
newQR() -> generarNumeroAleatorioConLongitud()	Genera un número aleatorio con una longitud aleatoria entre 5 y 15
registra()-> getInsert()	Al momento de que el visitante sea registrado, se almacenará su información en la BDV, y entonces insertará a un nuevo visitante.

## Diagrama buscaVis

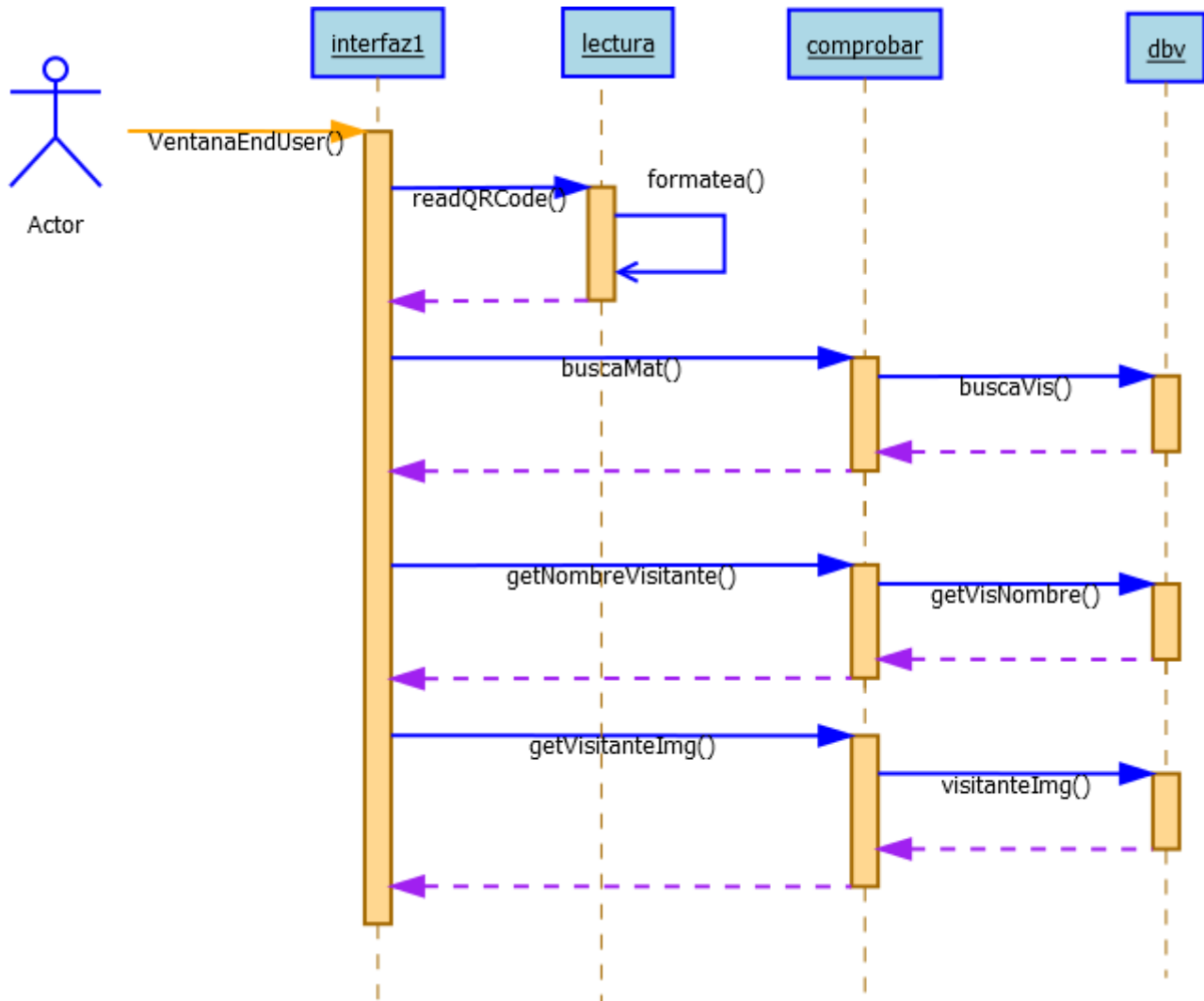


Diagrama	visEncontrado
Nos sirve para saber si el visitante que quiere ingresar al plantel ya ha sido registrado en la base de datos (dbv).	
Método	Descripción
VentanaEndUser()-> readQRCode()	Se conecta con el escáner, el cual lee un código QR. Se procesa el enlace y retorna el identificador.
readQRCode() -> formatea()	Formatea el enlace web obtenido, extrae el identificador almacenado dentro del enlace y retorna solo el identificador.
VentanaEndUser()-> buscaMat() -> buscaVis()	Del identificador obtenido por el escáner, comprueba si este existe en alguna de las bases de datos. De existir, retorna un Sting, indicando en qué DBV encontró el identificador.

<code>VentanaEndUser()-&gt; getNombreVisitante() - &gt; getVisNombre()</code>	Envía como parámetro el identificador, este se lo envía a las demás clases, el tipo de dato retornado es un Sting, el cual contiene el nombre completo de un visitante.
<code>VentanaEndUser()-&gt; getVisitanteImg() -&gt; visitanteImg()</code>	Envía por parámetro el identificador, este al final es envía a la DBV, en la cual realiza la búsqueda para obtener la identificación oficial almacenada en la base de datos.

## Definiciones, acrónimos y abreviaturas

- I. SCAM-QR:** Sistema de Control de Acceso Mediante QR.
- II. Campus:** Área de instalaciones universitarias donde se realizan actividades académicas y administrativas.
- III. Servidor:** Sistema informático que proporciona recursos y servicios a otros ordenadores a través de una red.
- IV. Base de Datos:** Conjunto organizado de datos almacenados electrónicamente, permitiendo su gestión y actualización.
- V. Normativas:** Reglas y directrices establecidas por una autoridad para regular comportamientos y acciones.
- VI. Políticas:** Normas que regulan las actividades y comportamiento dentro de la institución.
- VII. 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.
- VIII. 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.
- IX. 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.
- X. 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.
- XI. End User:** Usuarios finales, pertenecen a la UACM, los cuales utilizaran el programa, estos constan de estudiantes, docentes, personal administrativo, personal de limpieza, jardineros, personal de seguridad.
- XII. Visitante:** Usuario final, el cual no pertenece de ninguna manera al plantel educativo.
- XIII. DB:** Base de datos de la UACM.
- XIV. End Visit:** Usuarios temporales, los cuales buscan acceder por breve tiempo al plantel Cuauhtémoc, estos tendrán que hacer un registro con el Vigilante para tener control de su información y permanencia dentro del plantel. Para así obtener un código Qr temporal y permitir el acceso.
- XV. Usuario:** Consta del End User y Visitante.
- XVI. Usuario especial:** Usuario proporcionado por el área de sistemas, el cual, es utilizado para realizar consultas a la base de datos.
- XVII. Matricula:** Número de identificación único, proporcionado por la UACM.

## 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.