



# UParking

Universidad del Norte  
Dpto. De Ingeniería De Sistemas Y Computación  
Diseño de Software

18 de noviembre de 2022

**Profesor:** Wilson Nieto Bernal

## **Estudiantes:**

Carlos Alberto Otero Peña - 200121071  
Christian David Manga Arrazola - 200143579  
Paula Andrea Briceño Carrillo - 200143950  
Sebastián Enrique Perea Lopez - 200143589

## Índice

<b>1</b>	<b>Introducción</b>	<b>4</b>
<b>2</b>	<b>Objetivos</b>	<b>4</b>
2.1	Objetivo general . . . . .	4
2.2	Objetivos específicos . . . . .	4
<b>3</b>	<b>Metodología para el desarrollo</b>	<b>5</b>
<b>4</b>	<b>Marco conceptual</b>	<b>6</b>
<b>5</b>	<b>Cronograma del proyecto</b>	<b>8</b>
<b>6</b>	<b>Arquitectura de la solución</b>	<b>9</b>
<b>7</b>	<b>Modelos</b>	<b>10</b>
7.1	Casos de Uso . . . . .	10
7.2	Modelado de Costos . . . . .	11
7.3	Modelado de requerimientos . . . . .	12
7.3.1	Requerimientos generales . . . . .	12
7.3.2	Requerimientos de dos niveles . . . . .	13
7.4	Modelado de datos . . . . .	14
7.4.1	Modelo conceptual . . . . .	14
7.4.2	Modelo lógico . . . . .	15
7.5	Modelado de componentes . . . . .	16
7.6	Modelado de despliegue . . . . .	17
<b>8</b>	<b>Bases de Datos</b>	<b>18</b>

## Índice de figuras

1	Estimaciones del proyecto en términos de tiempo . . . . .	5
2	Cronograma . . . . .	8
3	Arquitectura de la solución . . . . .	9
4	Estimación de costos . . . . .	11
5	Requerimientos generales - simples . . . . .	12

6	Requerimiento de dos niveles . . . . .	13
7	Modelo conceptual . . . . .	14
8	Modelo lógico . . . . .	15
9	Modelo de componentes . . . . .	16
10	Modelo de despliegue . . . . .	17

# 1 Introducción

El proyecto tiene como propósito modelar, diseñar e implementar una aplicación móvil para agilizar el proceso de hacer uso de un servicio de parqueadero en cualquiera de sus actividades, localizar un estacionamiento y cancelar el costo del mismo, el área de funcionamiento corresponde a la ciudad de Barranquilla y su área metropolitana. Para ello, la aplicación debe articularse con los sistemas internos del cliente, y así obtener, un flujo de información confiable.

El usuario haciendo uso de Uparking podrá programar, a corto plazo, el destino, y asimismo, planificar la ruta óptima para evitar atascos vehiculares y largas filas de espera, siguiendo el mismo principio de ahorro de tiempo, podrá efectuar su pago por medio de la aplicación. A partir de los registros obtenidos, se realizará un análisis profundo para predecir la disponibilidad de los estacionamientos en las diferentes horas del día.

## 2 Objetivos

### 2.1 Objetivo general

Agilizar los procesos de localización y pago de un estacionamiento para los usuarios de los parqueaderos locales a través de una aplicación móvil.

### 2.2 Objetivos específicos

1. Determinar la disponibilidad en parqueos de un estacionamiento en específico.
2. Predecir la ocupación de los estacionamientos a determinadas horas del día.
3. Optimizar el pago del servicio de parqueaderos al integrar distintos métodos de pago.

### 3 Metodología para el desarrollo

Cascada, para realizar el proyecto es necesario definir fases que garanticen culminar cada etapa de manera exitosa con relación a los objetivos, evitando gastos innecesarios de recursos y tiempo.

Fases del proyecto	Entregables	Tiempo estimado	Responsable
Modelado de Requerimientos	Documento maestro con requerimientos y objetivos	1 semana	Tech Lead
Planeación	Definición de pasos a seguir para lograr objetivos	1 semana	Tech Lead
Documentación	-	2 semanas	Arquitecto
Implementación de Solución	API con sistema del cliente	1 mes	Software Developer
Implementación de sistema de pagos	Sistema de pago online		
Integración con app base	Integración con sistema del cliente		
Prueba de unidad	Caracterización de vulnerabilidades	1 mes	Tester / Software Developer
Prueba del sistema			
Operación	Aplicación movil base	2 semanas	Software Developer
Ingesta, transformación y análisis	Base de datos para el análisis de patrones en la ocupación	Recurrente	Cientifico de Datos

**Figura 1:** Estimaciones del proyecto en términos de tiempo

## 4 Marco conceptual

A continuación se describen los conceptos relacionados con el proyecto y la Ingeniería de Software:

1. **Ingeniería de Software** es una profesión que se ocupa de la cadena de valor de construcción y mantenimiento de sistemas de software. Permite dirigir y participar en equipos de desarrollo de software orientado a cualquier sector productivo, desde el diseño, la implementación y la gestión de arquitecturas de sistemas de información [1].
2. **Paradigmas de la ingeniería de software**
  - Ciclo de vida clásico.
  - Ciclo de vida clásico con prototipo.
  - El modelo de espiral.
  - Una combinación de estilos.
  - Prototipado puro.
  - Objetual.
3. Los **procesos de Software** son un conjunto de personas, estructuras de organización, reglas, políticas, actividades y sus procedimientos, componentes de software, metodologías, y herramientas utilizadas o creadas específicamente para definir, desarrollar, ofrecer un servicio, innovar y extender un producto de software [2].
4. **Sistemas de información** permiten articular diferentes áreas de conocimiento, permitiendo automatizar procesos operativos, apoyar el proceso de toma de decisiones al proporcionar información, y por consecuencia, facilitar el logro de ventajas competitivas a través de su implantación dentro de una organización [3].
  - (a) Sistemas transicionales: Reflejan un sistema en su comportamiento cotidiano.
  - (b) Sistemas de control de la gestión: son mecanismos utilizados por directivos y empleados para facilitar la consecución de los objetivos de la organización [4].
  - (c) Sistemas de apoyo a la toma de decisiones: son capaces de representar escenarios de posibles situaciones acordes a ciertos parámetros, investigaciones, pronósticos, o simulaciones.

5. **Ciclo de vida del Software** es una aproximación lógica a la adquisición, suministro, desarrollo, explotación y mantenimiento del software, correspondiente a la norma IEEE 1074 [5].

6. **Modelo de ciclo de vida** funciona como marco de referencia para la creación de un producto de software [5]. Contiene a detalle los procesos, actividades y tareas involucradas en el desarrollo, explotación y mantenimiento del mismo, y describe el desarrollo del proyecto desde la fase inicial a la final [6]. Está definido por la norma ISO 12207-1.

#### 7. **Ciclo de influencia de la arquitectura**

En la arquitectura inciden aspectos técnicos, administrativos y sociales, sus interacciones permiten aproximar los sistemas a las necesidades del proyecto, a su vez, la misma arquitectura influye en los factores antes mencionados, creando lo que se conoce como ciclo de influencias.

#### 8. **Estilos de arquitectura**

Los estilos de arquitectura corresponden a conjuntos de arquitecturas que comparten características similares, se detallan de manera lógica por lo que no deben incluir determinadas tecnologías. Entre las principales se encuentran:

- (a) N-Niveles
- (b) Microservicios
- (c) Big Data
- (d) Web - Cola - Trabajo
- (e) Basado en eventos
- (f) Big Compute

## 5 Cronograma del proyecto

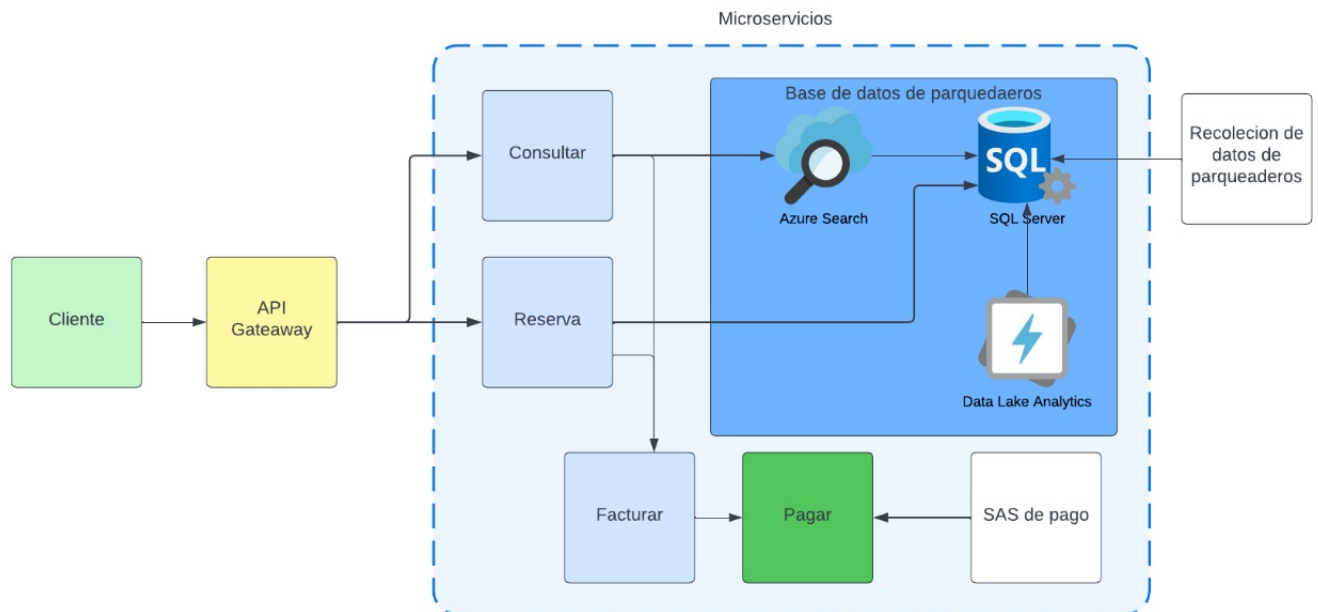
Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
PLANEACIÓN PROYECTO	7 días	vie 19/08/22	lun 29/08/22	
GESTIÓN REQUISITOS	7 días	mar 30/08/22	mié 7/09/22	1
DISEÑO	7 días	jue 8/09/22	vie 16/09/22	2
IMPLEMENTACIÓN	30 días	sáb 17/09/22	jue 27/10/22	3
Solución	15 días	sáb 17/09/22	jue 6/10/22	
Sistema de pago	8 días	vie 7/10/22	mar 18/10/22	5
Integración con APP base	7 días	mié 19/10/22	jue 27/10/22	6
PRUEBAS	7 días	vie 28/10/22	lun 7/11/22	4
Prueba de unidad	4 días	vie 28/10/22	mié 2/11/22	7
Prueba de sistema	3 días	jue 3/11/22	lun 7/11/22	9
DESPLIEGUE Y CIERRE	7 días	mar 8/11/22	mar 15/11/22	8
Ajustes	4 días	mar 8/11/22	vie 11/11/22	9
Puesta a punto	2 días	sáb 12/11/22	lun 14/11/22	12
Presentación y cierre	1 día	mar 15/11/22	mar 15/11/22	13

**Figura 2:** Cronograma



## 6 Arquitectura de la solución

La arquitectura implementada permite integrar múltiples servicios independientes, la comunicación entre ellos se da a través de API's gateways. Encontramos los servicios de acceso y registro, consulta, reserva y facturación. Los componentes de consulta y reserva acceden de manera directa a las bases de datos de los parqueaderos; La consulta implementa Azure Search y SQL Server, y la reserva ingresa únicamente a SQL server. Luego de la recolección de los datos se procede con un componente de analítica de datos para luego poder implementar un modelo de machine learning con la infraestructura proporcionada por Azure, y así, garantizar una funcionalidad a futuro que es la predicción de lugares disponibles dentro de un estacionamiento.

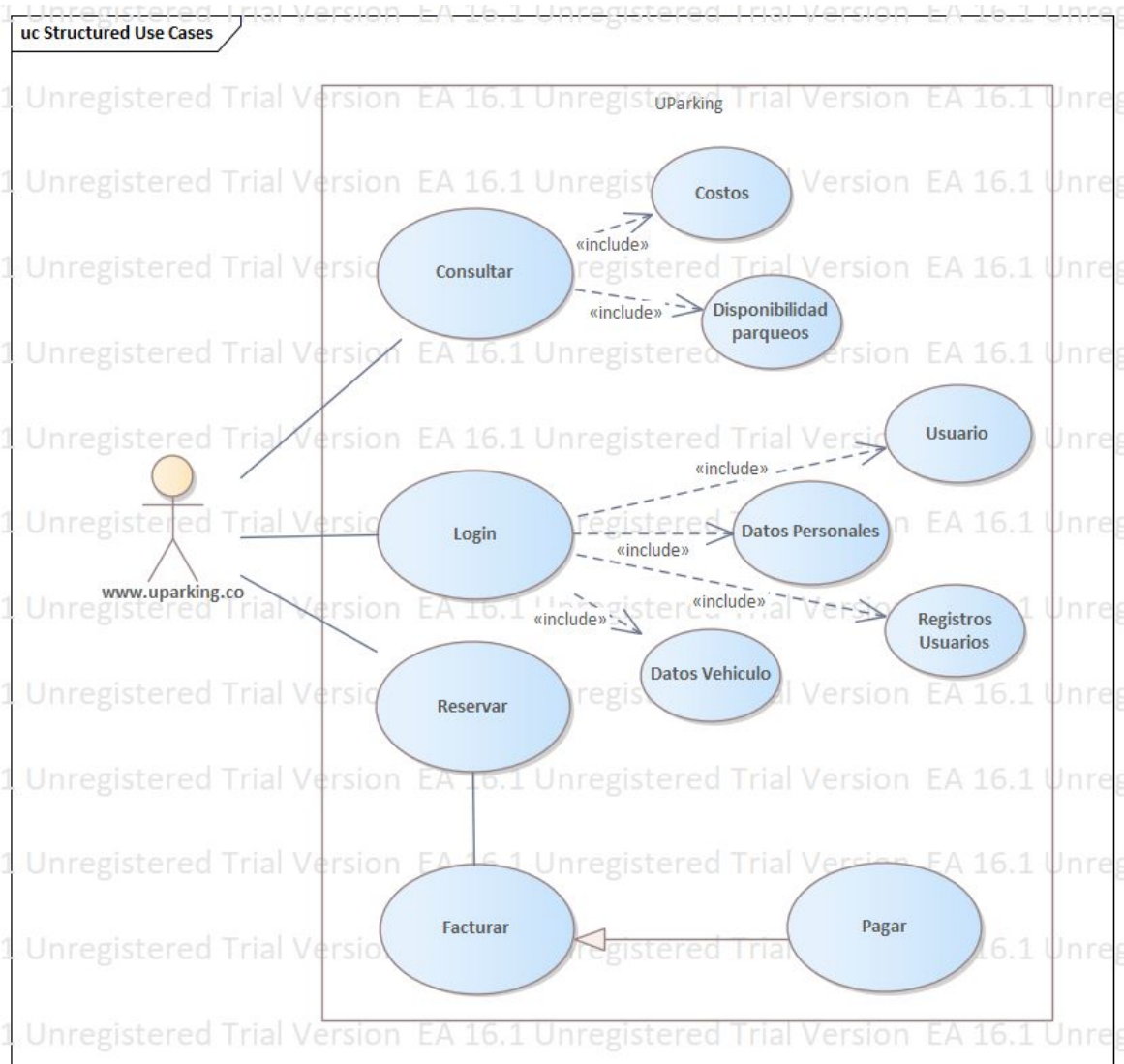


**Figura 3:** Arquitectura de la solución

## 7 Modelos

### 7.1 Casos de Uso

En el diagrama de los casos de uso se encuentran las principales actividades y/o servicios que el cliente de Uparking podrá realizar. Entre ellas, la consulta de los costos, por hora y fracción, y la disponibilidad de los espacios en estacionamientos; el ingreso a la plataforma donde se podrá registrar como un usuario con la posibilidad de añadir a un perfil única información sensible como sus datos personales y los datos de su vehículo; y por último, podrá efectuar el pago del servicio a través de una factura virtual generada por la aplicación.



## 7.2 Modelado de Costos

Se retratan a manera general los costos que el proyecto implica.

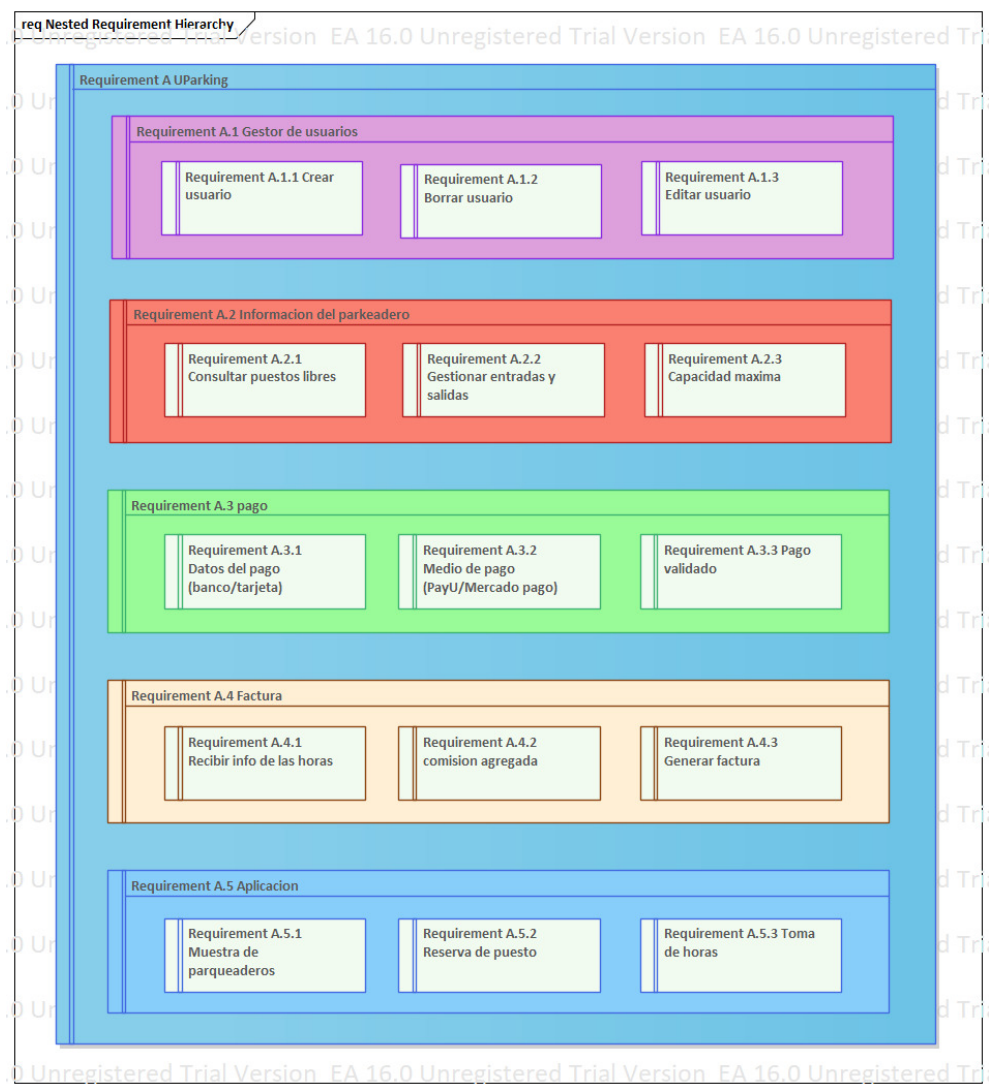
<b>PRESUPUESTO</b>		
<b>RUBROS</b>	<b>Subtotal</b>	<b>TOTAL</b>
Equipos	\$ 24,000,000.00	0
Talento humano	\$ 36,000,000.00	0
Tools; Software Deployment and Developed (PaaS)	\$ 2,500,000.00	0
Bibliografía (opcional)	\$ -	0
Servicios técnicos (instalaciones antes, después)	\$ 4,000,000.00	0
Publicidad y mercadeo (Mostrar, Vender)	\$ -	0
Trabajo de campo (encuestas, entrevistas, visualizar)	\$ 750,000.00	0
Gastos de Viajes (conferencias, rondas de negocios, Visitas tech)	\$ 5,600,000.00	0
Sub-total (..)	\$ 72,850,000.00	0
Costos Administración 10%-15%	\$ 7,285,000.00	0
Imprevistos 10%-15%	\$ 7,285,000.00	0
<b>Total</b>	<b>\$ 87,420,000.00</b>	<b>0</b>
Utilidad operacional	\$ 17,484,000.00	
<b>total con utilidad operacional</b>	<b>\$ 104,904,000.00</b>	

**Figura 4:** Estimación de costos

## 7.3 Modelado de requerimientos

### 7.3.1 Requerimientos generales

Para el desarrollo de la aplicación se precisa mantener la información de los parqueaderos (estacionamientos disponibles, costos de uso y capacidad). Los usuarios deben contar con las opciones disponibles para poder tomar una decisión que se adecúe a sus necesidades, por ello para agilizar el proceso de localización se implementa la posibilidad de reservar el estacionamiento.



**Figura 5:** Requerimientos generales - simples

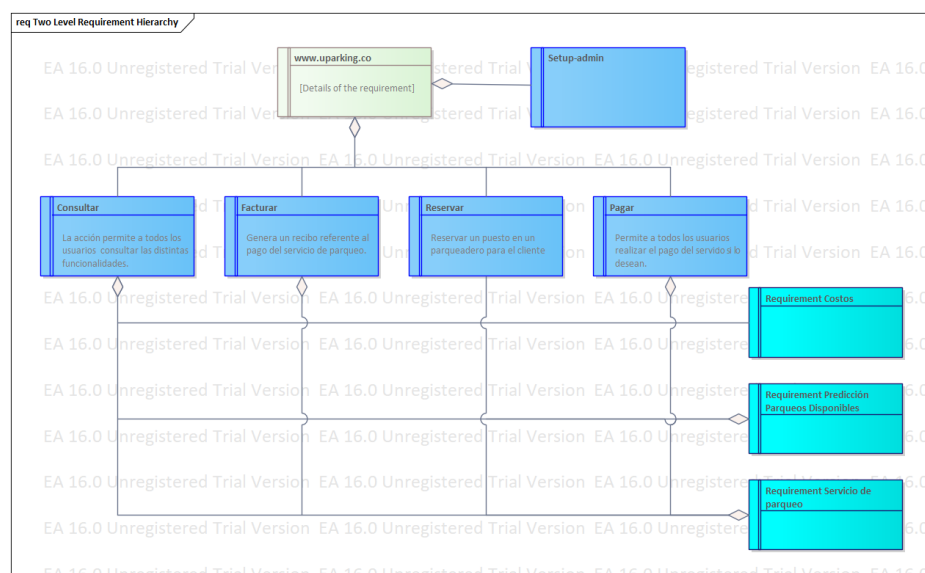
### 7.3.2 Requerimientos de dos niveles

UParking mejora la experiencia de llegar a un lugar sin tener que preocuparse por no encontrar estacionamiento o cancelar el disfrute del servicio. Dentro de los requerimientos generales encontramos:

- Consultas: Los usuarios podrán acceder para consultar las distintas funcionalidades de la plataforma.
- Facturas: Los usuarios podrán generar facturas virtuales.
- Reservas: Los usuarios podrán reservar sus espacios de parqueo dependiendo la disponibilidad y capacidad del parqueadero.
- Pagos: Los usuarios podrán efectuar su pago a través de la plataforma.

Los cuales necesitan de los siguientes requerimientos secundarios:

- Los **costos** asociados a los estacionamientos por horas y fracción.
- La disponibilidad de los espacio de parqueo a determinadas horas del día que corresponde al componente de **predicción**.
- El **servicio de parqueo** disponible.

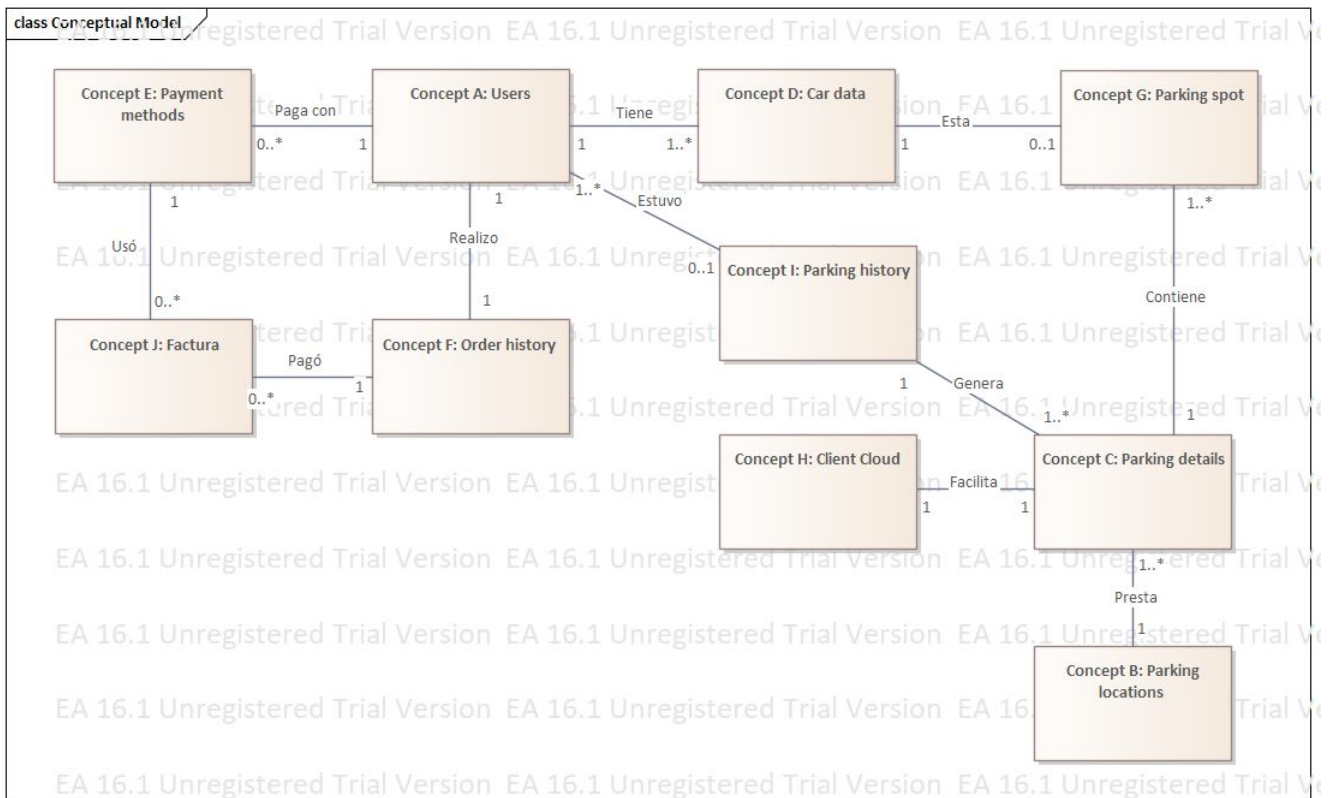


**Figura 6:** Requerimiento de dos niveles

## 7.4 Modelado de datos

### 7.4.1 Modelo conceptual

En este modelo se muestran los principales elementos que interactúan en la aplicación, es necesario que se retrate exclusivamente los conceptos principales de manera general porque así se garantiza una buena ejecución a partir del flujo lógico creado.



**Figura 7:** Modelo conceptual



### 7.4.2 Modelo lógico

Se describe a profundidad el alcance de las bases de datos, campos, entidades, propiedades y relaciones. Este modelo suele ser muy técnico para garantizar que todos los componentes se encuentren relacionados de manera adecuada y no haya fuga de datos o asociaciones incoherentes.

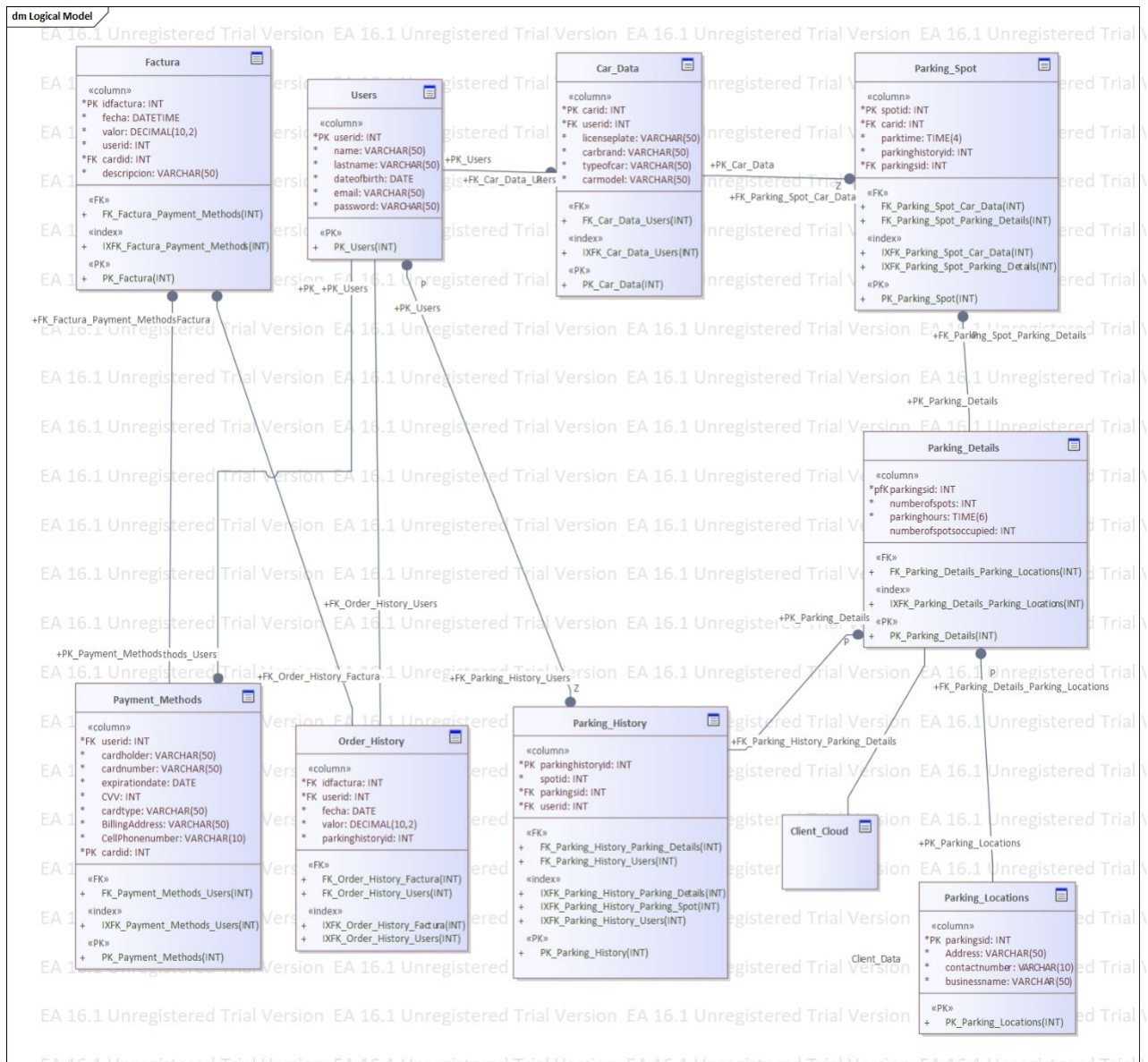
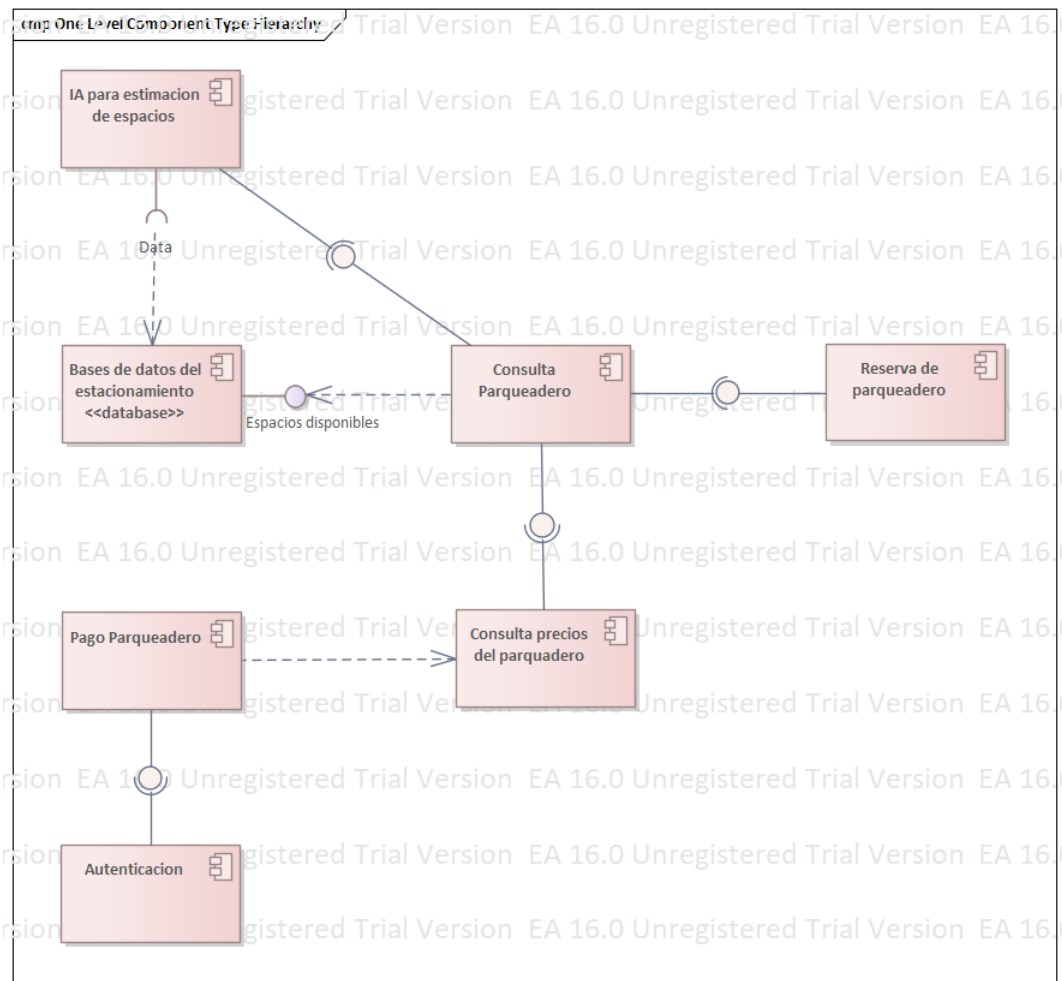


Figura 8: Modelo lógico

## 7.5 Modelado de componentes

El funcionamiento de la aplicación necesita de los componentes descritos a continuación. Al ingresar es necesario contar con la autenticación necesaria, una vez listo se puede proceder con el pago del parqueadero, sin embargo, ello depende del precio estipulado por el estacionamiento y la disponibilidad del mismo; En caso de reservar han de tenerse en cuenta todos los aspectos mencionados pero dependiendo principalmente de la disponibilidad. Para finalizar se implementa el componente de análisis y machine learning para crear modelos que sean capaces de predecir el comportamiento de la ocupación de los parqueaderos a determinadas horas del día.

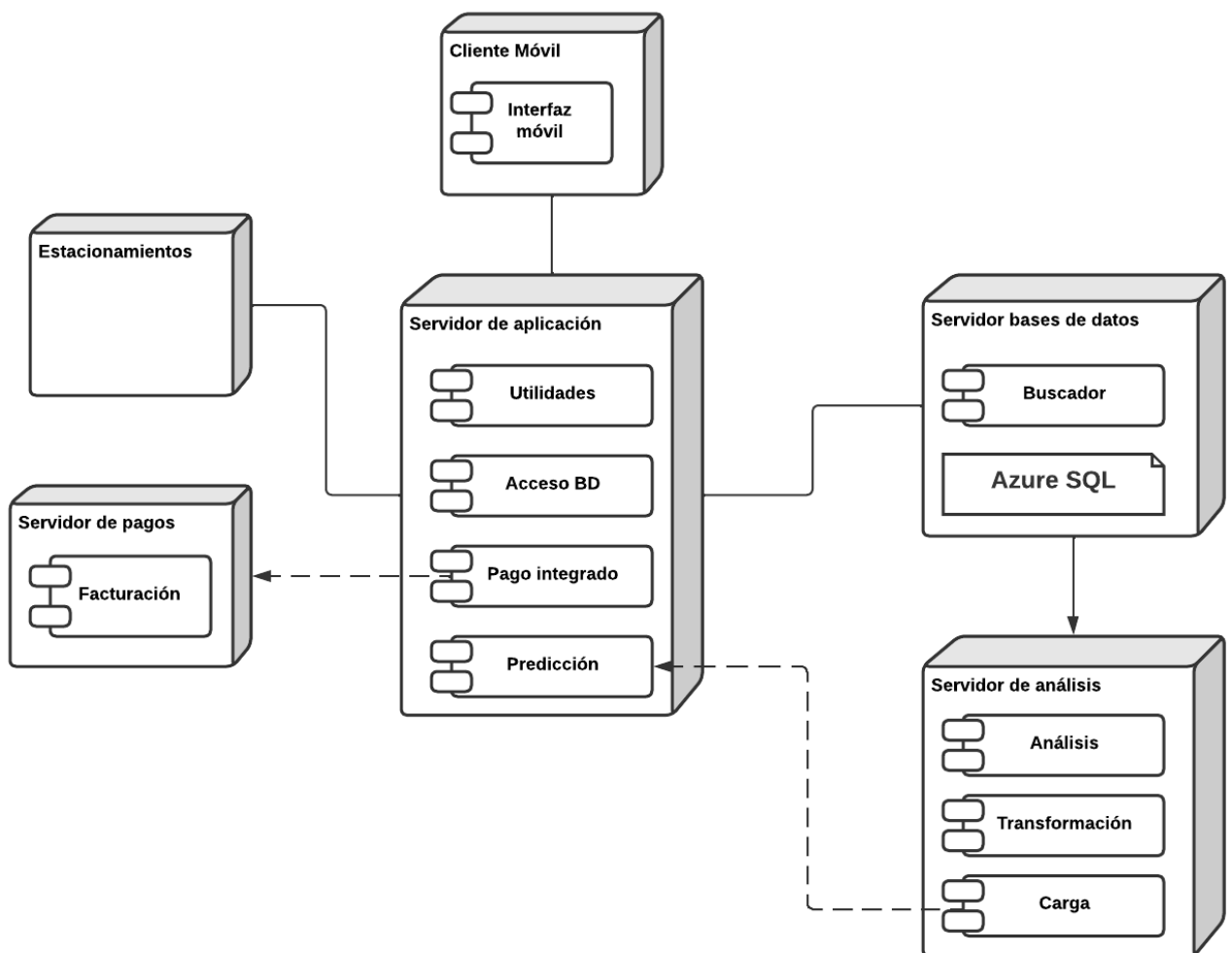


**Figura 9:** Modelo de componentes



## 7.6 Modelado de despliegue

La arquitectura requiere de los siguientes elementos para funcionar de manera adecuada, los componentes se encontrarán alojados en servidores virtuales a los que se tendrá acceso desde una plataforma móvil. La aplicación estará recibiendo información constantemente de los estacionamientos y de los usuarios, la información es alojada y analizada posteriormente transformada y suministrada al usuario de manera visual. La aplicación cuenta con una conexión directa a un servidor de pagos dependiendo del sistema administrativo del estacionamiento.



**Figura 10:** Modelo de despliegue

## 8 Bases de Datos

```
1 CREATE TABLE IF NOT EXISTS users(  
2     userid serial primary key,  
3     name VARCHAR(50) NOT NULL,  
4     lastname VARCHAR(50) NOT NULL,  
5     dateofbirth date NOT NULL,  
6     email VARCHAR(50) NOT NULL,  
7     password VARCHAR(50) NOT NULL  
8 );  
9  
10 CREATE TABLE IF NOT EXISTS factura(  
11     idfactura serial primary key,  
12     fecha timestamp NOT NULL,  
13     valor float NOT NULL,  
14     userid int NOT NULL,  
15     cardid int NOT NULL,  
16     descripcion VARCHAR(50) NOT NULL  
17  
18 );  
19  
20 CREATE TABLE IF NOT EXISTS car_data(  
21     carid serial primary key,  
22     userid int NOT NULL,  
23     licenseplate VARCHAR(50) NOT NULL,  
24     carbrand VARCHAR(50) NOT NULL,  
25     typeofcar VARCHAR(50) NOT NULL,  
26     carmodel VARCHAR(50) NOT NULL  
27 );  
28  
29 CREATE TABLE IF NOT EXISTS parking_spot(  
30     spotid int NOT NULL,  
31     carid int NOT NULL,  
32     parktime time NOT NULL,
```

```
33     parkinghistoryid int NOT NULL,
34     parkingsid int NOT NULL
35 );
36
37 CREATE TABLE IF NOT EXISTS payment_methods(
38     userid int NOT NULL,
39     cardholder VARCHAR(50) NOT NULL,
40     cardnumber int NOT NULL,
41     expirationdate date NOT NULL,
42     cvv int NOT NULL,
43     cardtype VARCHAR(50) NOT NULL,
44     billingaddress VARCHAR(50) NOT NULL,
45     cellphonenummer VARCHAR(10) NOT NULL,
46     cardid serial primary key
47 );
48
49 CREATE TABLE IF NOT EXISTS order_history(
50     idfactura int NOT NULL,
51     userid int NOT NULL,
52     fecha date NOT NULL,
53     valor float NOT NULL,
54     parkinghistoryid int NOT NULL
55 );
56
57 CREATE TABLE IF NOT EXISTS parking_history(
58     parkinghistoryid serial primary key,
59     spotid int NOT NULL,
60     userid int NOT NULL,
61     parkingsid int NOT NULL
62
63 );
64
65 CREATE TABLE IF NOT EXISTS parking_details(
66     parkingsid serial primary key,
67     numberofspots int NOT NULL,
```

```
68     numberofspotsoccupied int NOT NULL ,
69     parkinghours time NOT NULL
70 );
71
72 CREATE TABLE IF NOT EXISTS parking_locations(
73     parkingsid serial primary key ,
74     address VARCHAR(50) NOT NULL ,
75     contactnumber VARCHAR(10) NOT NULL ,
76     businessname VARCHAR(50) NOT NULL
77 );
78 CREATE TABLE IF NOT EXISTS client_cloud(
79
80 );
81
82 ALTER TABLE factura
83     ADD FOREIGN KEY (cardid) REFERENCES Payment_Methods ...
84         (cardid);
85
86 ALTER TABLE car_data
87     ADD FOREIGN KEY (userid) REFERENCES users (userid);
88
89
90 ALTER TABLE parking_spot
91     ADD FOREIGN KEY (carid) REFERENCES car_data (carid),
92     ADD FOREIGN KEY (parkingsid) REFERENCES ...
93         parking_details (parkingsid);
94
95 ALTER TABLE payment_methods
96     ADD FOREIGN KEY (userid) REFERENCES users (userid);
97
98
99 ALTER TABLE order_history
```

```
100     ADD FOREIGN KEY (idfactura) REFERENCES factura ...  
        (idfactura),  
101     ADD FOREIGN KEY (userid) REFERENCES users (userid);  
102  
103  
104 ALTER TABLE parking_history  
105     ADD FOREIGN KEY (parkingsid) REFERENCES ...  
        parking_details (parkingsid),  
106     ADD FOREIGN KEY (userid) REFERENCES users (userid);
```

## Referencias

- [1] P. G. Colombiano, “Ingeniería de software virtual,” <https://www.poli.edu.co/profesional/ingenieria-de-software-virtual>, Jun. 2016, accessed: 2022-8-30.
- [2] Software Guru, “Procesos de software,” <https://sg.com.mx/revista/1/procesos-software>, accessed: 2022-8-30.
- [3] A. H. Trasobares, “Los sistemas de información: Evolución y desarrollo.”
- [4] J. Berbel, “Sistemas de control de gestión: qué son y qué tipos de usos tienen,” <https://edem.eu/sistemas-de-control-de-gestion-que-son-y-que-tipos-de-usos-tienen/>, Nov. 2021, accessed: 2022-8-30.
- [5] M. N. Moreno García, “Modelos de proceso del software,” in *Tema 2 - Análisis de Sistemas*, Universidad de Salamanca, Ed. Universidad de Salamanca.
- [6] Autónoma, Universidad, D. Carmen, A. A. Canepa, S. Christian, and E. García González, “COMPARATIVAS DE LOS MODELOS DE CICLO DE VIDA,” Tech. Rep.