

# FACULTAD DE INGENIERIA INGENIERIA DE SISTEMAS

#### DESARROLLO BASADO EN PLATAFORMAS

## ACTIVIDAD I

AUTORES: CAMILA MARTINEZ CRISTIAN NOVOA

DOCENTE: JUAN CARLOS MARTINEZ

# BOGOTÁ D.C

Línea de Atención al Usuario: 593 30 04 • Línea Nacional: 01 8000 936670 www.uniminuto.edu

- Reximo el ciclo de vida de construcción de un programa El ciclo de vida de construcción de un programa generalmente sigue estas possos
  - 1. Análisis del problema: Se define el problema que resolverá con el programa
  - 2. Diseño: Se planifica la estructura del programa, algoritmos y diagramas de flub
  - 3. Codificación: Se escribe el código en un lenguage de programación
  - 4. Pruebas y depuvación: Se verifica que el programa funcione correctamente y se corrigen errores
  - 5. Implementación: Se despliega el programa en un entorno real
  - 6. Mantenimiento y actualización: Se reculian mejoras y convecciones a lo largo del tiempo
- b. Aspectos del análisis de un problema
  - El analois de un problema incluye
    - 1. Identificación del problema: Se comprende la necesidad o el désafio a resolvér
    - 2. Recolección de requisitos: Se determinan las necesidades del uxorio y las restricciones del sistema
    - 3. Definición de entradas y salidas: Se identifican los datos de entrada y los resultados esperados
    - 4. Limitaciones y restricciones: Se consideran aspectos como tiempo de ejecución, recursos y seguridad
    - 5. Alternativas de solución: Se evalúan diferentes enfoques pava vexoluer el problema

- c. Etapos del proceso de solución de problemas Consta de los aquientes etapos
  - 4. Comprensión del problema: Analizar en detalle el problema y sus requerimientos
  - z. Plany reación de la solución: Diseñar el alapritimo o estrategia poura' verduer el problema
  - 3. Implementación de la solución: Escubir el códo o desarrollar la solución planteada
  - 4. Pruebas y validación : Verificar que la solución funcione ... correctamente en distintos escerarios
  - 5. Documentación y montenimiento: Registrou el tuncionamiento del sistema y mejorar su vendimiento
- d. Elementos a entrepr a un cliente

tos elementos que se delben de entregar a un cliente en un 11 proyecto de software incluye

- 1. Código tuente: Si el cliente requere acceso al código del brodrama
- z. Hanual de ususuro : Documento con instrucciones para usar el sistema
- 3. Monual técnico : Información sobre la arquitectura y el montenimiento del software
- 4. Documentación de requisitos y diseño: Especificaciones técnicas y funcionales del sistema
- 5. Producto final: El software instalado y listo para su uso
- 6. Capacidación : Formación para los usuavios finales si es
- 7. Soporte técnico y garantía: Asistencia en caso de ervores o mejovas

ALL PROPERTY OF A PROPERTY OF

Debe haber un sistema de seguridad pora autenticor a los uscorros, como el uso de layetas y PINs.  • El programa debe manegar posibles errores, como tondos insuficientes o fallos en la conexión con la base de dalos  Requermiento  • Seguridad: El programa debe govantizar la seguridad de las transacciones y la privacidad de la información de los usarios  • Disponibilidad: Los asperos automáticas deben estar aperativas las el carror del día, los 7 días de la semana y tácil de usar para los clientes	Cliente	El banco que desea implementar el programa
Rz: El programa debe permilir a la vasoria consulta el saldo de sus cuentas  Mundo del problema de la universación de las cuentas de las universación de las cuentas despandos y la científicación del titular  El programa debe tener acceso a la base de datas tenaciones y actualizar los saldos despues de coda retiro  Debe haber un sulama de seguridad pora autenticad a las vetiro  Debe haber un sulama de seguridad pora autenticad a la universación de las prodes inspicientes o fallas en la conexión con la taxe de autors  el programa detre manegar posibles errores como tendos inspicientes o fallas en la conexión con la taxe de autors  Requermiento de las tronsacciones y la privacidad de la información de los unavivos  Deponibilidad: la alercia automáticas deben estar apendinas las 31 hanos del día los 7 días de la semana y taxin de usar pora los dentes  Pendimiento: El programa debe procesar los transacciones de monero rapidas y eficiente para evitar largas esperas	Ουνακύ	los clientes del banco que utilizaván los cajeros automaticos pava realizat transacciones
Hundo del problema de las usualizar conscer la información de las cuentas de las usualizar como el número de cuenta, el salab dispansioles y la ribertificación del titular dispansioles y la ribertificar del titular del banco para userticar la autenticidad de las transacciones y actualizar los salabs despues de cada retiro.  Debe haber un sistema de seguridad para autenticar a la usualizar como el uso de tayetas y PINs.  El programa detre mangar possibles errores como tondos insificientes o fallas en la conexión con la base de datas.  Requermento de las transacciones y la pricipadad de la inflormación de las transacciones y la pricipadad de la inflormación de los usuarras.  Cupansidiada: la interpar del capero debe ser influtiva y Tácil de usar para los clientes.  Pendimiento: El programa debe procesar las transaccion de monera rapidad y eficiente para evitar largus esperas.	Requerimento puncional	R1: El programa debe permitir a los usuarios retiro
de la visivios como el número de cuenta, el saldo despandole y la identificación del titular  El programa debe tener acceso a la base de datas del borno paro verticor la autenticidad de las transacciones y actualizar los saldos despues de cada retiro  Debe haber un sistema de seguridad para autenticar a los uscarios, como el uso de tayetas y 1918s  El programa debe manegar pasibles errores como tondos inspicientos o tallas en la conexión con la base de datos  Requermento es seguridad: El programa debe quantizar la seguridad de las transacciones y la priusicidad de la información de los usarvios  Espandolidad: las aperos automálicas deben estar aperalicas las el harias del día, los 7 días de la semana y tácil de usar para los clientes  Pendimiento: El programa debe procesar las transacciones de monero rapidos y eficiente para evitar largas esperas	200 DOST 97	Rz: El programa debe permitir a los usuarios consulta el salab de sus cuentas
Debe haber un silama de seguridad para autenticar a la uscarios, como el uso de largetas y PINS  • El programa debe manegar posibles erroves, como tondos insificientos o tallos en la conexión con la base de datos  Requenmento e Seguridad: El programa debe gorantizar la seguridad de las transacciones y la privacidad de la inflavmación de las usarvios  • Disponibilidad: las asteros culomáticas deben estar aperativas las el horas del día, las 7 días de la semuna  • Usabilidad: las interpas del capero debe ser intuitiva y tácil de usar para los clientes  • Rendimiento: El programa debe procesar las transaccios nes de manera rapada y eficiente para evitar largas esperas		· Es necesorio conocer la información de las cuentas de los usuarios como el número de cuenta, el saldo disponible y la identificación del titular
Chapanilitad: La interpar del da capero de la semuna  Condita inspirana delse manejar possibles erroves, como tondo inspirana de la conexión con la base de dalos  Requermiento  Seguridad: El programa delse aprodizar la seguridad de la información de las inspiracións y la privacidad de la información de los uxarvios  Cosponibilidad: La aleros automálicas deban estar aperalicas los est interpor del día, los 7 días de la semana  Cospolidad: La interpor del capero debe ser intuitiva y tocil de usar para los clientes  Pendimiento: El programa delse procesar las transacciones de manero rapida y eficiente para evitar largus esperas	Chilop	· El programa debe tener acceso a la base de datas del banco para verticar la autenticidad de las transacciones y actualizar los salobs despues de cada retiro
Requeremento e Seguridad: El programa debe garantizar la seguridad no funcional de las Ivanisaciones y la privacidad de la información de los uxarvios.  Disponibilidad: los asperos automáticas deben estav aperativas las 24 horas del día, los 7 días de la semana.  Osabilidad: la interpar del capero debe ser intultiva y fácil de usar para los clientes.  Rendimiento: El programa debe procesar las transacciones de manera rapidas y eficiente para evitar largas esperos.	T ( 12 K + 12	- Debe haber un sistema de seguridad pora autenticor a los uxorios, como el uxo de tayetas y PINS
de los uxarvios  Disponibilidad : los asperos outomáticos deben estav operativos los 24 horos del dío, los 7 días de la semana  Usabilidad: la interfaz del capero debe ser intultiva y Tácil de usar para los clientes  Pendimiento: El programa debe procesar las transaccio nes de manera rapida y eficiente para evitar largas esperos	17-11-11	· El programa delse manegar possibles errores, como tondos insuficientes o fallos en la conexión con la bose de datos
<ul> <li>Disponibilidad : las coleros automáticas deben estav operativos los 34 horos del día, los 7 días de la semana.</li> <li>Usabilidad : las interpas del cajero debe ser intuitiva y Tácil de usav para los clientes</li> <li>Pendimiento : El programa debe procesar los transacciones de monero rapidas y eficiente para evitar largas esperas</li> </ul>	Requeremento no funcional	· Seguridad: El programa debe garantizar la seguridad de la información de la uxariat
<ul> <li>Usabliobd: La interfaz del cajero debe ser intuliua.</li> <li>y Tácil de usar para los clientes</li> <li>Pendimiento: El programa debe procesar los transacciones de monero rapida y eficiente para evitar largas esperas</li> </ul>		
experior and an analysis and a		[11] 시간 [2] [2] [2] [2] [2] [2] [2] [2] [2] [2]
Eepman   1904		
	<b>C</b> 1 1 1 1	· Pendimiento: El programa debe procesor los transacción nes de monera vapicha y eficiente para evitar largas esperas

	Nombre	K1: Realizar un depásito en una cuerta
	Rexumen	Permite a un uxavio depositar una cantidad de dinero en su cuenta barroavia
	Entrodas	Número de cuenta Contidod a depositar
	Resultado	El salob de la cuenta ha sub incrementado con la contrabal depositada
2 .		
	Nombre	Rz Realizar un relivo de una cuenta
	1 Resumen	Permite a un usuario retinar una cantidad de dinero de su aventa bancaria
	Enlvados	Número de cuenta Controbol a retivar
	Reultado	El soldo de la cuenta ha sido disminuido con la contidad retirada siempire que haya fondos sificientes
	<b>,</b> les per	
	Nombve	Rs: Consultar el saldo de una cuenta
	Resumen	Permite a un uxorro Consultar el solob actual de su cuenta boncaria
	Entrodas	Número de aventa
	Revillado	Se muestra el saldo adual de la cuenta al luxuario
	THAT!	EN AND CHE FERCY INFLORMANCE TO DESCRIPTION

Ta	rea 3 Requerimien	TOS
	Nombre	R1: Determinar et tipo de triargulo
	Reumen	Permite determinar si un triángulo es equilátero. Vásceles o exalero basado en las longitudes de sus lados
0	Entradas	longilludes de las tres lados del triángulo
	Resultado	Se mustra el lipo de livargulo Cequilátero, isosceles o exalero?
2.	Requerimien	
	Nombre	Rz: Calcular el parimetro del triángulo
	Rexumen	Permite calcular el perimetro de un triángulo sumando los longitudes de sus tres locals
	Enlvachs	longilludes de los 3 londos del triángulo
	Resultada	Se muestra el perimetro del triangulo
3,	Requerimient	on surprise the partial of the same of the
	Nombre	R3: Calcular el area del Inargula
	Resumen	Permite calculor el area de un triangulo utilizando la formula de Heron
	Entrados	torgitudes de las 3 lados del triangula
	Reullado	Se muestra el area del trióngulo

H,	Tavea 4	to a special section of the section	
	140 140	Nombre	
	Entiabol	Triángulo	La entiable principal que representa un triangula con tres lados y tres ángulas
	Entidad	lado	Representa cools una de los tres loclos del trangula, con una longillo especifica

Punto de reflexión CODE para si no identificamos bien las entidades del mondo

Keprexenta cacha uno de las tres orgulos del

trianquilo con una medida en grados

Pademos enfrentas serios desafios en el diser y desarvollo del programa. Un mal distro locado en entrodes incorrectas puede resillar una estructura confusa y dificil de mantener, lo que complica la implementación de funcionalidades. Además podificionas implementar conacteristicas que no se aliman con los necesidades recellos del produteros reales del problema

Punto de reflexión à Cómo dead y si se trata ejediuamente de una entidad y no solo de una cavadenitica de una entidad ya identificada.P

Es impostante considerar varios factores, primero evaluar a el concepto puede existir de monera indepondiente y si tiene sus propias características y comportamientos. Si es así es probable que sea una enticad. Segundo, anatizar a el concepto tiene relaciones significations où otras antidades, lo que también signere

#### Tarea 5

Entidad

Anaplo

Claxe: Cuentaboncaria

Otudinta	Valves Posibles
numeroCuenta	String Closhi picador de la cuenta)
tholar	String (Vombre del titular de la cuenta)
Saldo	Double Coalob actual de la cuenta
Fedanperlura	String (fecha de apertura de la cienta)
Estado	String (Estado de la cuenta : Activa, Inactiva, blapeada

• 6	ograma.	UML	æ	Cuento	kancar
	Cuer	itakono	ρινα		
	- numero - Litula: - Saldo - Fechar - estado	Cuenta: 1000 1000 1000 1000 1000 1000 1000 10	ng ng ble s slu	ving	

Alvibuta	Valorestoubles
I I mite Sobveguo	bouble
Codnerer	bouble

· biognoma uma Cuenta Connente

	Cue	nta Co	wente
- 1	limite	Sobreg	ivo: bouble bouble
-	taxaln	leres :	bouble

· Clase : Cunta Alburos

1	Atributo	Valores Possibles
I	taxaInteres Anual	bouble CTaxa de interés anual)
	sald Minimo	boulde (Souldo minimo requesido pouva cuitou congo

· Diagrama UML de Cuenta Alhorros

II.	ntalhorics
taxalnt	eresArval & Double
- saldol	limmo 3 double

Class COT

F	Alviboto	Valores Parables
	númeroCDT.	String (Identificador Unica del CDT
X	conteunlation	bouble CHonb invalido en el CDT)
	Plazo	int (Plazo en meses del clot)
	Taxalnlerer	Double (Tous de Merér del CHT)

· biograma UML de COT

CDT	A
- numeroCDT: String	
- mantalnuertido : bouble	-
- Plazo: int	
- taxalnleren; double	

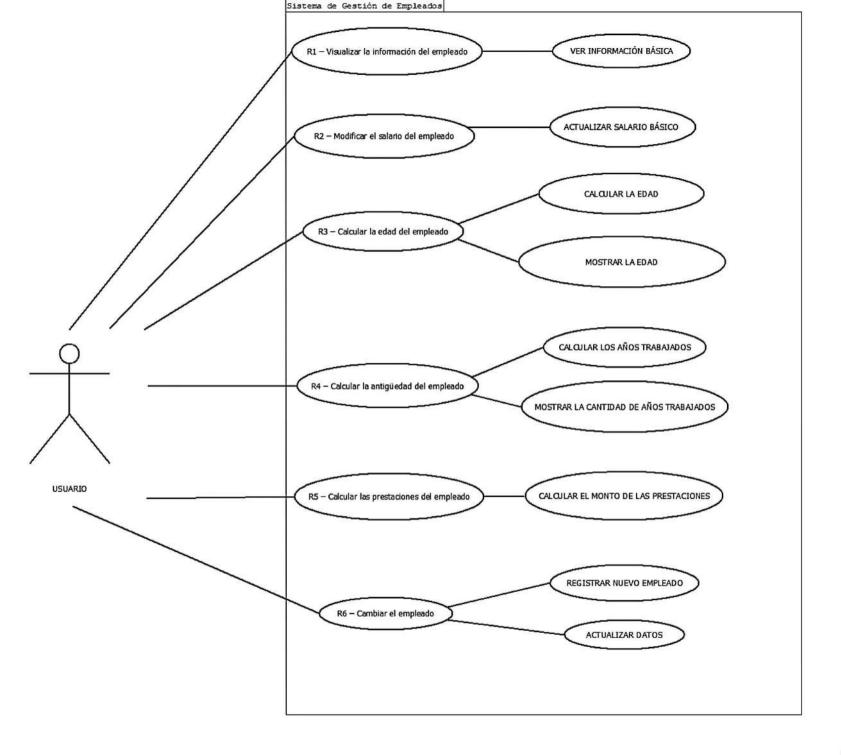
· Diagrama UML de Mes

-	Me2		
_	numerolles sint		
-	nombrelles: Strug		

· Claze : Hes

Alributo	Valores Possibles
numeroHes	int (Número del mes en la simulación)
nombrelles	String (Nombre del mes)

Tarea 6 J Conclusiones: El alprilmo presentado para viapr en el metro de Pará, ounque proporciona una guía general carece de aficiente precisión y detalle para evitar ambigüedodes. For ejemplo, no expecífica cómo identificar correctamente las eslaciones en el majos, como distinguir entre multiples líneas que podricín llevor al destino, o que hocer en coso de transferencias entre líneas. Ademais, supone que el usuario tiene un conocimiento previo del sistema de metro y utiliza su "sentico común" para interpretar los instrucciones, lo que puede llevar diferentes interpretaciones y errores. Para que un algoritmo sea efectivo, debe ser lo suficientemente detallado y clato para cualque perora k. Ejemplo 1: El empleado · Enunciado del problema ? Se requiere una aplicación que permita gestionar la información de un empleado, proporcionardo funcionalidades para visualizar sus datos, modificar su salario, calcular la edad, antiguedad y prestaciones, así como cambor el empleado registrado en el sistema · Requerimientos funcionales 1,81 - Visualizar la información del empleado 2.Rz-Modificar el salario del empleado 3. Rs - Coulculor, la echod del empleado 4.84 - Calculor la antigüedad del empleado 5. Rs - Calculor los prestaciones del empleado 6. R6 - Combios el empleado special property and the of the and priorbeid : blat and



Punto m M

Clase: Empleado

Atributos (variables)

nambre (String): Nombre del empleado
apellido (String): Apellido del empleado
genero (String): Genero del empleado
fecha-nacimiento (Date): Fecha de nacimiento del empleado
Fecha-ingreso (Date): Fecha de ingreso a la empresa
salario (barbe): Salario basico del empleado
imagen (String): Ruta o URL de la imagen del empleado

# N. Punto N

Idea de proyecto 1: Sistema de gestión de Tareas y prioridades

Requerimiento	Nambre : RegistrarTarea
functional 1	Reximen : Permite al uxorio registror una nueva torea con su descripción, prioridad y tiempo estimado
maiczkad m Talo słacia	Entrados: Descripción (String), Prioridad (Int), Tiampo
	Resultado: La torea se añode a la lista de toreas pendentes
Requerimiento Funcional 2	Nombre: AsignorPrioridad
funcional 2	Rezumen: Permite al coustic asignar o modificar la prioridad de una torea existente
_ leb settin	_ Entradas = 10 de la torea (INT), nueva prioridad (int)
	Reviltado: La prioriched de la torea se advalisa
Requerimients Funcional 3	Nombre: Calculartiempo Total
tuncional 5	Rexumen : Calcula el tiempo total extimado pova completor todos las toreas pendientes
Tipling webs	Entracto: Wingura
	Rexultados Devuelue el trempo total estimodo Chaulte)

Requerimiento	Nombre: Optimizar Taveas
Funcional 4	Resumen & Utiliza un algoritmo pora reorganizar las
	Entroobs Ningura
	Resultado: La lista de toveas se veordena poura aptimador la productividad

# Idea de Proyecto: 2 : Sistema de recomendación de Rulas de viaje

Requerimento	Nombre	Registron Destino
Functional 1	Reumen	Permite al uzorio registror un nuevo destino con su nombre, coordenados y costo asociado
-do-object	Entradas	Nombre (Strug), Coordenactos (boulde, louide), Costo (boulde)
abalwanii a	Resultado	El destro se añade a la lista de destinos disponibles
Requerimiento Funcional 2	Nombre	Calcula-Ruta Optima
Funcional 2	Rewmen	Utiliza un algoritmo (como Dijkstra) pora calculor la ruta más eficiente entre dos destiros
11/2	Entradas	bestino Inicial (String), bestino Final (String)
Spart (da) t	Samuel All Cont	Devuelue la rula optima y el cost total
Reperimiento	Nombre	Mostrov Ruta
Funcional 3	Resumen	Muestra la ruta calculada en un formato Legible para el uxurio
varidon e	Entrodas	Ruta (List (String)
	Resultado	la ruta se muestra en la interfaz del uxorio
Requerimento	Nombre	Actualizar Costo
Functional 4	Rexumen	Permile al uxorio advalizar el costo axociodo a un destino
every don't	Enlucios	Nombre del Destino (String), Nevo Costo (Double)
	Rexultada	El costo del destino se actualiza en la



1.1 Describa y justifique el problema, indicando porque se trata de un problema soluble (referencia pág. 5):

Problema	Manejo de reservas de un avión con capacidad limitada y	
	diferentes clases de sillas	
Cliente	Empresa de aerolíneas que necesita un sistema eficiente	
	para gestionar reservas de pasajeros.	
Usuario	Personal de la aerolínea encargado de la gestión de	
	reservas y pasajeros	
Requerimiento	-Asignar sillas a pasajeros según preferencias de clase y	
funcional	posición.	
	-Consultar, buscar y eliminar reservas.	
	-Calcular porcentaje de ocupación y valores de ventas.	
Mundo del problema	Gestión de reservas en un avión con 50 sillas (8 ejecutivas	
	y 42 económicas), considerando preferencias de los	
	pasajeros y optimización del espacio	
Requerimiento no	-Sistema debe ser fácil de usar y eficiente.	
funcional -Debe manejar datos de pasajeros de manera seg		
	-Debe proporcionar cálculos precisos y rápidos.	

## Justificación del problema soluble:

- 1. **Cliente y Usuario:** El problema es relevante para una empresa de aerolíneas que necesita gestionar reservas de manera eficiente. El sistema beneficiará al personal de la aerolínea al simplificar y agilizar el proceso de reservas.
- 2. **Requerimientos Funcionales:** Los requisitos funcionales están claramente definidos, lo que permite desarrollar un sistema que cumpla con las necesidades específicas de la aerolínea. Las operaciones de asignación, consulta, eliminación y búsqueda son tareas manejables con un programa bien diseñado.
- 3. **Mundo del Problema:** El contexto del problema es limitado y bien definido (50 sillas con clases específicas), lo que facilita la creación de un sistema que pueda manejar estas restricciones de manera efectiva.
- 4. **Requerimientos No Funcionales:** Los requisitos no funcionales, como la usabilidad, seguridad y precisión, son alcanzables con las tecnologías y metodologías de desarrollo adecuadas.



ALTA

CALIDAD

CALIDAD

O SERVING

1.2 Señale mínimo 4 requerimientos funcionales indicando el tipo de dato que existen en el lenguaje de implementación para las variables que serán utilizadas en las entradas y salidas del requerimiento

	Nombre	Asignar una silla a un pasajero
	Resumen	Asignar una silla a un pasajero según sus preferencias de clase y posición.
REQUERIMIENTO FUNCIONAL 1	Entradas	- Nombre del pasajero (String) - Cédula del pasajero (String) - Preferencia de clase (String: "ejecutiva" o "económica") - Preferencia de posición (String: "ventana", "pasillo", "centro")
	Salidas	- Confirmación de asignación (String: "Asignación exitosa" o "No hay sillas disponibles") - Detalles de la silla asignada (String: "Clase: ejecutiva, Posición: ventana, Silla: 1A")
	Nombre	Consultar una reserva
	Resumen	Consultar los detalles de una reserva utilizando la cédula del pasajero.
	Entradas	- Cédula del pasajero (String)
REQUERIMIETO FUNCIONAL 2	Salidas	- Detalles de la reserva (String: "Nombre: Juan Pérez, Cédula: 123456789, Clase: económica, Posición: pasillo, Silla: 10B") - Mensaje de error si no se encuentra la reserva (String: "Reserva no encontrada")
	Nombre	Eliminar reserva
	Resumen	Eliminar una reserva existente utilizando la cédula del pasajero.
REQUERIMIENTO	Pérez, Cédula: 123456789, Clase: económica Posición: pasillo, Silla: 10B") - Mensaje de error si no se encuentra la reserva (String: "Reserva no encontrada")  Nombre Eliminar reserva  Resumen Eliminar una reserva existente utilizando la cédula del pasajero.  Entradas - Cédula del pasajero (String)	
FUNCIONAL 3	Salidas	<ul> <li>Confirmación de eliminación (String: "Reserva eliminada exitosamente")</li> <li>Mensaje de error si no se encuentra la reserva (String: "Reserva no encontrada")</li> </ul>
	Nombre	Calcular el porcentaje de ocupación del avión
REQUERIMIENTO FUNCIONAL 4	Resumen	Calcular el porcentaje de sillas ocupadas en el avión.



Nombre	Calcular el porcentaje de ocupación del avión
Entradas	- Número de sillas ocupadas (Int) - Número total de sillas (Int: 50)
Salidas	- Porcentaje de ocupación (Float: 75.0%)

2.1 Identifique las entidades del mundo problema, mínimo 3 (consulta como referencia la pág. 16, J. Villalobos) RUBBY CASALLAS GUTIERREZ, JORGE ALBERTO VILLALOBOS SALCEDO, "Fundamentos de Programación: Aprendizaje Activo Basado en Casos" En: México 2006. Ed:Pearson Education ISBN: 970-26-0846-5

ENTIDADES (Las que necesitas)	NOMBRE	DESCRIPCIÓN
1	Pagaiero	Representa a un pasajero que realiza una reserva. Contiene información como nombre y cédula.
2	Silla	Representa una silla en el avión. Contiene información sobre su clase (ejecutiva o económica) y posición (ventana, pasillo, centro).
3	Reserva	Representa la reserva de una silla por un pasajero. Contiene información sobre el pasajero, la silla asignada y detalles de la reserva.

2.2 Señale las características de las entidades descritas:

## Entidad #1: Pasajero

NOMBRE DE LA ENTIDAD #1	VALORES POSIBLES (Rango aceptable)	TIPO DE DATO (C++) Y EXPLICACIÓN
Nombre	Cualquier cadena de caracteres (ej: "Juan Pérez")	std::string - Para almacenar nombres completos con espacios y caracteres especiales.
Cédula	Números enteros positivos (ej: 1234567890)	int o long long - Dependiendo del tamaño de la cédula, se usa int para valores pequeños o long long para valores más grandes.



#### Entidad #2: Silla

NOMBRE DE LA ENTIDAD #2	VALORES POSIBLES (Rango aceptable)	TIPO DE DATO (C++) Y EXPLICACIÓN
Clase	"Ejecutiva" o "Económica"	std::string - Para almacenar la clase de la silla como una cadena de texto.
Posición	"Ventana", "Pasillo", "Centro"	std::string - Para almacenar la posición de la silla como una cadena de texto.

#### Entidad #3: Reserva

NOMBRE DE LA ENTIDAD #3	VALORES POSIBLES (Rango aceptable)	TIPO DE DATO (C++) Y EXPLICACIÓN
Pasajero	Objeto de tipo Pasajero	Pasajero - Se usa un objeto de la clase Pasajero para representar al pasajero que realiza la reserva.
Silla	Objeto de tipo Silla	Silla - Se usa un objeto de la clase Silla para representar la silla asignada.
Detalles de la reserva	Cualquier cadena de caracteres (ej: "Reserva confirmada")	std::string - Para almacenar información adicional sobre la reserva.

2.3 Establezca las relaciones entre las entidades de forma lógica, en un esquema gráfico, estableciendo las entidades y las relaciones (investiga: Diagrama E/R) (Pág. 21-22):

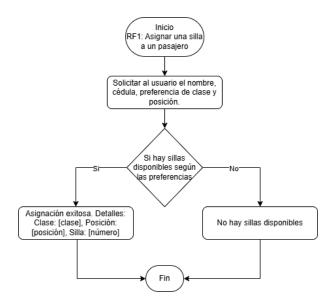




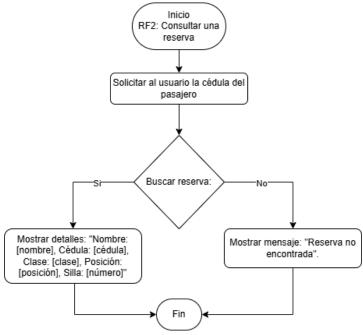


# 2.4 Por cada requerimiento funcional desarrollado en el numeral 1.2, crea un diagrama de flujo:

# Requerimiento Funcional 1



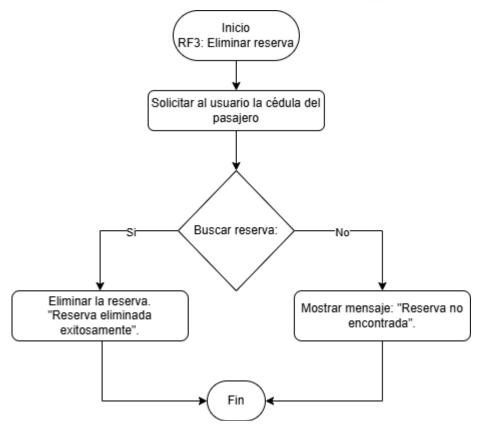
#### Requerimiento Funcional 2







#### Requerimiento Funcional 3







## Requerimiento Funcional 4

