

raada en tres niveles de logro:

Criterio	No alcanzado (0-4)	Alcanzado parcialmente (5-7)	Alcanzado completamente (8-10)
1. Uso de Git	No se utiliza Git o se usa incorrectamente sin seguimiento adecuado de versiones.	Se utiliza Git para el control de versiones pero hay errores ocasionales en el uso de ramas o commits.	Uso experto de Git con commits claros y precisos, manejo adecuado de ramas y colaboración efectiva.
2. Pruebas unitarias JUnit	No se implementan pruebas unitarias o son insuficientes para verificar la funcionalidad.	Se implementan pruebas pero no cubren todos los casos críticos o tienen errores.	Pruebas unitarias completas y correctas que garantizan la cobertura de todos los aspectos críticos del código.
3. Documentación JavaDoc	Documentación inexistente o muy escasa.	Documentación incompleta o con errores en la descripción de métodos y clases.	Documentación completa y detallada que facilita la comprensión y mantenimiento del código.
4. Refactorización del código	No se aplica refactorización o se aplica de manera que no mejora la legibilidad o estructura del código.	Se realiza refactorización, pero es parcial y no abarca todos los componentes necesarios.	Refactorización efectiva que mejora significativamente la legibilidad y mantenimiento del código.
5. Diagrama de flujo, pseudocódigo y tabla de decisión	Ausencia de estos elementos o presentados de forma incorrecta e inútil.	Elementos presentes pero con errores o falta de detalle que impide su completa utilidad.	Presentación clara y correcta de al menos un diagrama de flujo, pseudocódigo y tabla de decisión que aportan valor al entendimiento del proyecto.
6. Grafo de flujo y complejidad ciclomática	No se presenta grafo de flujo o se calcula incorrectamente la complejidad ciclomática.	Grafo de flujo presente pero con errores menores en el cálculo de complejidad o en la especificación de caminos.	Correcta elaboración de grafo de flujo con un cálculo preciso de la complejidad ciclomática y especificación adecuada de caminos y pruebas.
7. Documentación de requisitos	Documentación de requisitos funcionales y no funcionales inexistente o inadecuada.	Documentación incompleta o poco clara de los requisitos funcionales y no funcionales.	Documentación completa y detallada de todos los requisitos funcionales y no funcionales, proporcionando claridad y dirección al proyecto.
8. Diagramas UML	Falta de diagramas UML o diagramas incorrectos e inútiles.	Diagramas UML presentes pero incompletos o con errores significativos.	Presentación completa y correcta de diagramas UML (de clases, de casos de uso, etc.) que reflejan adecuadamente la estructura y el funcionamiento de la aplicación.

Los 4 primeros requisitos son de código.

Los 4 siguientes tienen que introducirse en la memoria mediante imágenes y explicaciones de lo que realizan.

Se ha realizado refactorización de código y Javadoc.

Se ha realizado pseudocódigo,diagrama de flujo y tabla de decisión

No se ha podido llegar a realizar el punto 6

Se ha realizado el punto 7 y 8

Índice

Integrantes del Grupo: David,Jaime,Ian y Iván.

Información de lo que ha realizado cada participante del grupo cada día.

Dia 13/05/2024

- Todos hemos trabajado en el entidad-relación.
- Ian y Jaime han utilizado las 2 ultimas horas para actualizar el tablero Kanban.

Dia 14/05/2024

- Ian ha hecho el paso a tablas y ha empezado la normalización.
- Jaime ha hecho la especificación de la API HTTP y la base del backend .
- Ivan y David han hecho pantallas del frontend Android.

Dia 15/05/2024

- Jaime ha hecho el CRUD de películas en servidor.
- David ha hecho el CRUD de usuarios en servidor.
- Ian ha estado revisando el ddl porque había que añadir algo al ddl hecho por David.
- Ivan ha modificado el ddl hecho por David y pasarlo a Oracle.

Dia 16/05/2024

- Ivan ha estado añadiendo unos cambios al apartado preferencias en Android.
- David ha estado haciendo la segunda pantalla del apartado de Android.
- Ian ha estado haciendo la función anyadir_carrito y el procedimiento procecimiento_login.
- Jaime ha hecho el login en HTML y ha estado revisando el servidor.

Dia 17/05/2024

- Iván hizo el pseudocódigo y el diagrama de flujo de inicio de sesión y también la tabla de decisión.
- Jaime ha estado haciendo el PL/SQL
- David ha hecho el CRUD de Clientes en HTML.
- Ian ha estado haciendo el CRUD de clientes en JavaScript.

Dia 20/05/2024

- Jaime ha hecho el CRUD de clientes y está haciendo el de Películas.
- Ivan ha modificado el pseudocódigo y ha añadido partes a los requisitos funcionales y no funcionales.
- David está haciendo la conexión del servidor con Android Studio.
- Ivan está haciendo los requisitos funcionales y no funcionales de la aplicación.

Dia 21/05/2024

- Ian a finalizado el CRUD de clientes y está haciendo el de Películas.
- Ivan ha modificado el pseudocódigo y ha añadido partes a los requisitos funcionales y no funcionales.
- David ha conectado el servidor para poder iniciar sesión en móvil conectado al servidor.
- Jaime ha hecho más funciones y procedimientos.
- Día 22/05/2024
- Ivan ha hecho el Diagrama UML de todo el proyecto y luego de cada package para poder verse mejor y entenderse más.

- David ha estado haciendo modificaciones al proyecto de android.
- Ian ha estado acabando el CRUD de películas en JavaScript.

Dia 23/05/2024

- Ian ha terminado JavaScript y ha comentado el código en visual studio code
- David ha hecho el Javadoc en Servidor
- Jaime ha terminado unas cosas de android
- Ivan ha hecho el Javadoc de Android

Requisitos Funcionales y No Funcionales.

Requisitos Funcionales:

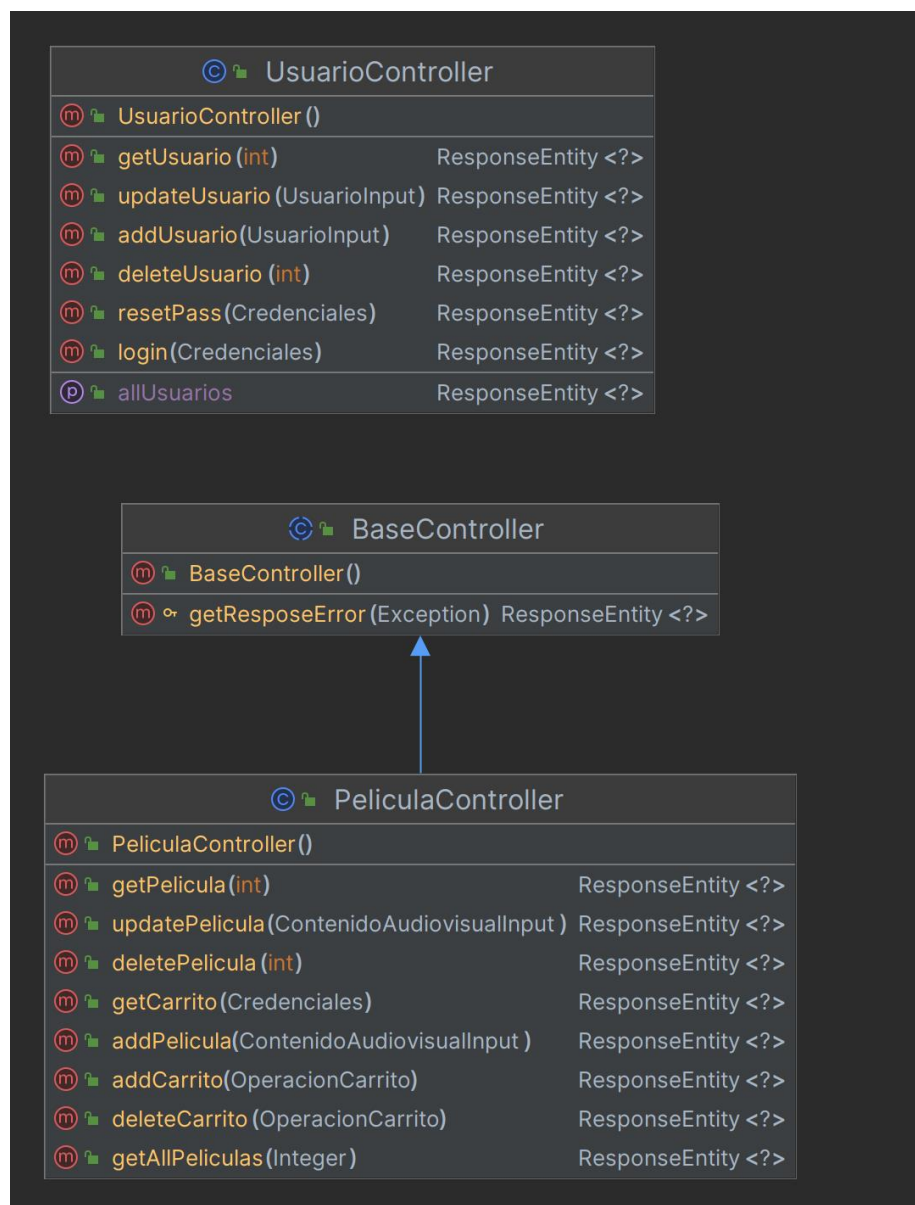
- Hacer autenticación
- Crear Cuenta
- Recuperar Contraseña
- Luego de autenticar Mostrar las películas
- Seleccionar película y verla al detalle toda su información
- Poder votar la película y clasificarla en un ranking de 0 a 5
- Alquilar la película mediante un carrito.
- Visualizar al detalle la información de la compra
- Un apartado preferencias donde cambiar el idioma y el tema de la aplicación.

Requisitos no funcionales:

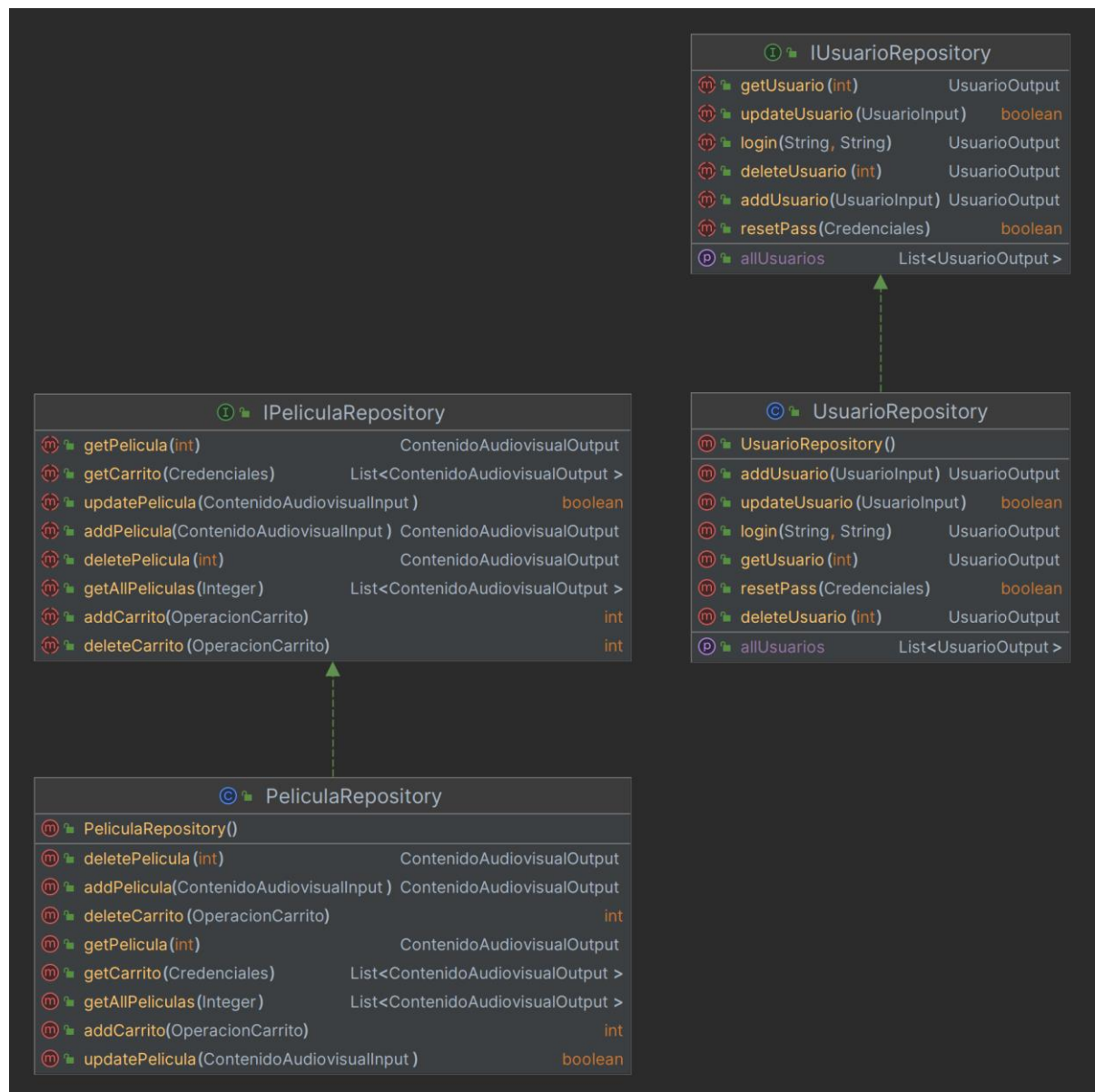
- La interfaz es intuitiva y fácil de usar.
- Los tiempos de respuesta son rápidos en un dispositivo móvil y a una velocidad aceptable al ser usados en el emulador de Android Studio
- Toda la información sensible se manda mediante un método Post encriptado.
- La aplicación estará disponible para el 99 % de los dispositivos android porque trabaja con Android Oreo.
- Soporta varios idiomas para poder ser utilizada por varias personas del mundo.
- Es sencillo de mantener gracias a la gran documentación del código realizada mediante Javadoc.

Aquí está la tabla de decisión de si un usuario tiene cuenta o no que debe hacer

Condición	1	2	3	4
Tiene cuenta	0	0	1	1
No tiene cuenta	1	0	0	1
Acciones				
Crear cuenta	x			x
Iniciar sesión		x	x	



Este es el UML del package controller

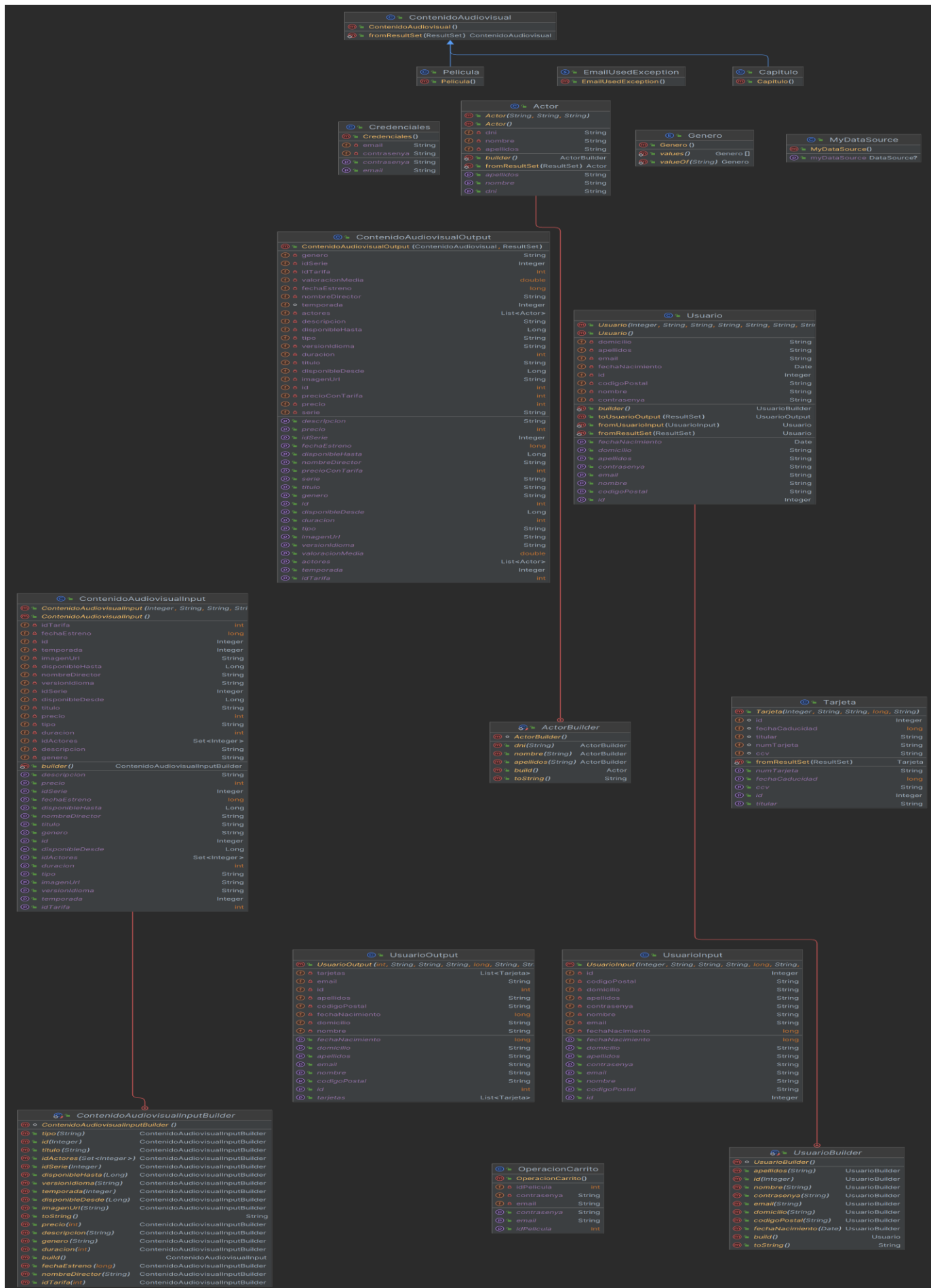


Este es el UML del repository

© 📁 UsuarioService		
Ⓜ 📁	UsuarioService()	
Ⓜ 📁	login(String, String)	UsuarioOutput
Ⓜ 📁	addUsuario(UsuarioInput)	UsuarioOutput
Ⓜ 📁	updateUsuario(UsuarioInput)	boolean
Ⓜ 📁	deleteUsuario(int)	UsuarioOutput
Ⓜ 📁	resetPass(Credenciales)	boolean
Ⓜ 📁	getUsuario(int)	UsuarioOutput
Ⓟ 📁	allUsuarios	List<UsuarioOutput >

© 📁 PeliculaService		
Ⓜ 📁	PeliculaService()	
Ⓜ 📁	deletePelicula(int)	ContenidoAudiovisualOutput
Ⓜ 📁	getCarrito(Credenciales)	List<ContenidoAudiovisualOutput >
Ⓜ 📁	updatePelicula(ContenidoAudiovisualInput)	boolean
Ⓜ 📁	addCarrito(OperacionCarrito)	int
Ⓜ 📁	addPelicula(ContenidoAudiovisualInput)	ContenidoAudiovisualOutput
Ⓜ 📁	deleteCarrito(OperacionCarrito)	int
Ⓜ 📁	getPelicula(int)	ContenidoAudiovisualOutput
Ⓜ 📁	getAllPeliculas(Integer)	List<ContenidoAudiovisualOutput >

Este es el UML del apartado Service



Y este que hay arriba es el de Model


Aquí está el PSEINT nuestro el código y el resultado

```
Algoritmo CrearCuentaUsuario
  Definir nombre, apellido, email, contrasena, domicilio, codigoPostal Como Cadena
  Escribir '=== Crear Cuenta de Usuario ==='
  Repetir
    | Escribir 'Ingrese su nombre:'
    | Leer nombre
    | Si nombre='' Entonces
    | | Escribir 'El nombre no puede estar vacío.'
    | | FinSi
  Hasta Que nombre!=''
  Repetir
    | Escribir 'Ingrese su apellido:'
    | Leer apellido
    | Si apellido='' Entonces
    | | Escribir 'El apellido no puede estar vacío.'
    | | FinSi
  Hasta Que apellido!=''
  Repetir
    | Escribir 'Ingrese su email:'
    | Leer email
    | Si email='' Entonces
    | | Escribir 'El email no puede estar vacío.'
    | | FinSi
  Hasta Que email!=''
  Repetir
    | Escribir 'Ingrese su contraseña:'
    | Leer contrasena
    | Si Longitud(contrasena)<8 Entonces
    | | Escribir 'La contraseña debe tener al menos 8 caracteres.'
    | | FinSi
  Hasta Que Longitud(contrasena)>=8
  Repetir
    | Escribir 'Ingrese su domicilio:'
    | Leer domicilio
    | Si domicilio='' Entonces
    | | Escribir 'El domicilio no puede estar vacío.'
    | | FinSi
  Hasta Que domicilio!=''
  Repetir
    | Escribir 'Ingrese su código postal:'
    | Leer codigoPostal
    | Si codigoPostal='' Entonces
    | | Escribir 'El código postal no puede estar vacío.'
    | | FinSi
  Hasta Que codigoPostal!=''
  Escribir ''
  Escribir '=== Confirmación de Datos ==='
```

```

42         Si codigoPostal='' Entonces
43             Escribir 'El código postal no puede estar vacío.'
44         FinSi
45     Hasta Que codigoPostal#''
46     Escribir ''
47     Escribir '=== Confirmación de Datos ==='
48     Escribir 'Nombre: ', nombre
49     Escribir 'Apellido: ', apellido
50     Escribir 'Email: ', email
51     Escribir 'Domicilio: ', domicilio
52     Escribir 'Código Postal: ', codigoPostal
53     Escribir ''
54     Escribir 'Usuario creado exitosamente.'
55 FinAlgoritmo
56

```

 PSeInt - Ejecutando proceso CREARCUENTAUSUARIO

*** Ejecución Iniciada. ***

=== Crear Cuenta de Usuario ===

Ingrese su nombre:

> Ivan

Ingrese su apellido:

> montesinosm

Ingrese su email:

> ivan@gmail.com

Ingrese su contraseña:

> 12344523

Ingrese su domicilio:

> fsdsfd dfd

Ingrese su código postal:

> 2342342

=== Confirmación de Datos ===

Nombre: Ivan

Apellido: montesinosm

Email: ivan@gmail.com

Domicilio: fsdsfd dfd

Código Postal: 2342342

Usuario creado exitosamente.

*** Fiección Finalizada ***

Este es el diagrama de flujo

