



Universidad  
Católica del Norte

# Informe 2:

# Taller 1

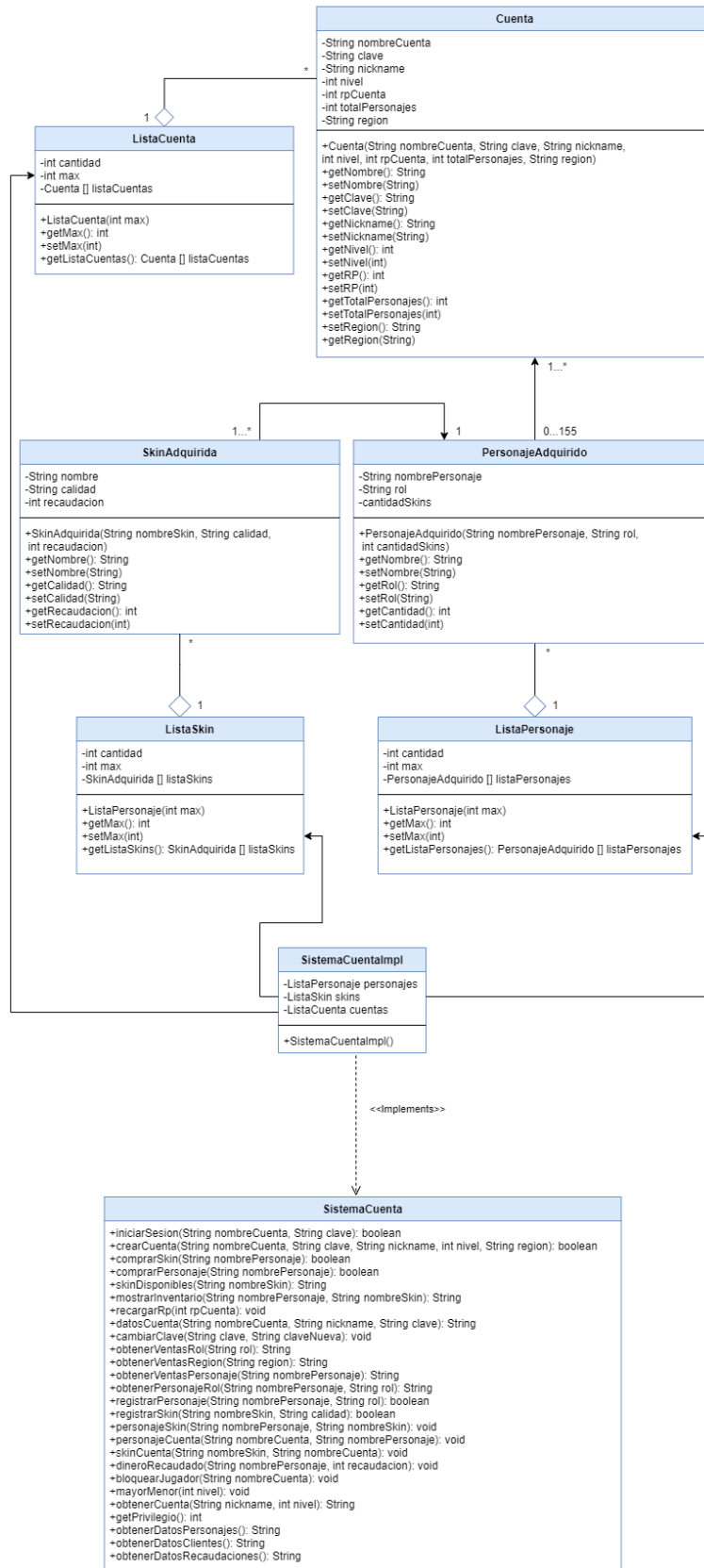
Asignatura: Programación Avanzada

Nombre profesor: Juan Torres

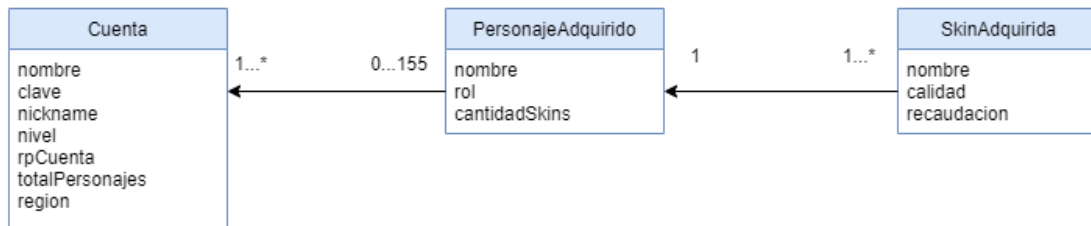
Integrantes: Matias Morales 20.168.337-8/Nicolas Peralta 21.004.634-8

Fecha: 31/10/2021

# DIAGRAMA DE CLASE



## MODELO DEL DOMINIO



## CONTRATOS:

### RF 2 (INICIO DE SESIÓN):

Operación	iniciarSesion(): boolean Parámetros de entrada: nombreCuenta, clave
Descripción	Contrato para iniciar la cuenta
Pre-condición	-Que este registrado
Post-condición	No hay

Operación	crearCuenta(): boolean Parámetros de entrada: nombreCuenta, clave, nickname, nivel, region
Descripción	Con los parámetros agregados permiten crear una nueva cuenta
Pre-condición	-Que esta cuenta aún no esté registrada
Post-condición	-Crear una nueva cuenta

### RF 3 (MENÚ CLIENTE):

Operación	comprarSkin(): boolean
Descripción	Contrato permite comprar una skin
Pre-condición	-Exista la skin en el sistema -Que exista el personaje del que pertenezca la skin -Que la cuenta no posea la skin -Tener RP suficiente para comprarla
Post-condición	-Agregar la skin a la cuenta

Operación	comprarPersonaje(): boolean
Descripción	Contrato permite comprar un personaje
Pre-condición	-Que exista el personaje en el sistema -Que la cuenta no posea al personaje -Tener RP suficiente para comprarlo
Post-condición	-Agregar un personaje a la cuenta

Operación	skinDisponibles(): String
Descripción	Muestra las skins que el cliente no poseé
Pre-condición	-Que hayan skins disponibles
Post-condición	No hay

Operación	personajesDisponibles(): String
Descripción	Muestra los personajes que el cliente no poseé
Pre-condición	-Que el cliente exista
Post-condición	No hay

Operación	mostrarInventario(): String Parámetros de entrada: nombrePersonaje, nombreSkin
Descripción	Muestra todos los personajes que tiene el cliente con las skins
Pre-condición	-Que hayan skins disponibles
Post-condición	No hay

Operación	recargarRp(): void
Descripción	Recarga el rp de la cuenta
Pre-condición	-Tener dinero para añadir a la cuenta
Post-condición	-Mayor cantidad de RP

Operación	datosCuenta(): String
Descripción	Muestra los datos de la cuenta
Pre-condición	-Cuenta registrada
Post-condición	No hay

Operación	cambiarClave(): void
Descripción	Permite cambiar la clave de la cuenta
Pre-condición	-Que exista una cuenta -Que este registrado
Post-condición	-Clave nueva agregada

**RF 4 (MENÚ ADMIN):**

Operación	obtenerVentasRol(): String
Descripción	Obtiene la recaudación de ventas por rol
Pre-condición	-Que exista una cuenta -Que este registrado
Post-condición	No hay

Operación	obtenerVentasRegion(): String
Descripción	Obtiene la recaudación de ventas por región
Pre-condición	-Tener el registro de region
Post-condición	No hay

Operación	obtenerVentasPersonaje(): String
Descripción	Obtiene la recaudación de ventas por personaje
Pre-condición	-Que exista el personaje
Post-condición	No hay

Operación	obtenerPersonajeRol(): String
Descripción	Obtiene cantidad de personajes por cada rol que haya
Pre-condición	-Que exista el personaje
Post-condición	No hay

Operación	registrarPersonaje(): boolean
Descripción	Permite ingresar datos para registrar al personaje
Pre-condición	-Que el personaje no esté registrado -Que no sobrepase la cantidad de personajes establecido (155)
Post-condición	-Existe un personaje más registrado

Operación	registrarSkin(): boolean
Descripción	Permite ingresar datos para registrar las skins de un personaje
Pre-condición	-Que la skin no esté registrada

Post-condición	-Existe una skin nueva para algún personaje
----------------	---

Operación	addPersonaje(): void Parámetros de entrada: nombre, rol
Descripción	Permite añadir un personaje al sistema
Pre-condición	-Que no exista el personaje -Que la lista de personajes no esté llena
Post-condición	-Se añade un personaje al sistema y se aumenta la cantidad de personajes que posee el sistema

Operación	addSkin(): void Parámetros de entrada: nombreA, calidadA, nombreP
Descripción	Permite añadir una skin a un personaje del sistema
Pre-condición	-Que la lista de aspectos de el personaje no esté llena -Que la skin a ingresar no se encuentre en el personaje -Que el personaje exista
Post-condición	-Se añade una skin a un personaje y se aumenta la cantidad de aspectos de ese personaje

Operación	addPersonajeCuenta(): void Parámetros de entrada: nombreP, nombreC
Descripción	Permite añadir un personaje a una cuenta
Pre-condición	-Que la lista de personajes poseídos no esté al límite -Que el cliente no posea al personaje -Que exista el personaje -Que exista el cliente
Post-condición	-Se añade un personaje a la lista de personajes poseídos del cliente. -La cantidad de personajes poseídos por el cliente aumenta.

Operación	addSkinCuenta(): void Parámetros de entrada: nombreP, nombreA, nombreC
Descripción	Permite añadir una skin a un personaje poseído por un cliente
Pre-condición	-Que el cliente exista

	-Que la skin exista -Que el personaje exista -Que el cliente posea al personaje -Que el personaje poseído no posea la skin
Post-condición	-Se añade una skin a un personaje poseído por un cliente, como también aumenta la cantidad de skins poseídas por el personaje poseído.

Operación	addRecaudacion(): void Parámetros de entrada: nombrePersonaje, recaudación
Descripción	Revisa si existe el personaje para ingresar el RP recaudado
Pre-condición	-Que el personaje exista
Post-condición	-Asociar el RP de un personaje

Operación	bloquearJugador(): void Parámetros de entrada: nombreCuenta
Descripción	No permite que un jugador pueda ingresar o registrarse con el mismo nombre de cuenta
Pre-condición	-Que la cuenta exista
Post-condición	-No poder volver a registrarse o entrar

Operación	mayorMenor(): void Parámetros de entrada: nivel
Descripción	Ordena de mayor a menor el nivel de cuentas
Pre-condición	-Que exista una cuenta
Post-condición	-Asociar el RP de un personaje

Operación	obtenerCuentas(): String Parámetros de entrada: nickname, nivel
Descripción	Obtiene el nickname de la cuenta y su nivel
Pre-condición	-Que exista una cuenta
Post-condición	No hay

Operación	getPrivilegio(): int Parámetros de entrada:
Descripción	Obtiene el número de privilegio de un cliente para iniciar sesión
Pre-condición	-Debe de existir la cuenta/cliente
Post-condición	No hay

Operación	obtenerDatosPersonajes(): String
-----------	----------------------------------

	Parámetros de entrada:
Descripción	Obtiene los datos que se encuentran en el archivo "personajes.txt"
Pre-condición	-Debe de existir el archivo de texto "personajes.txt"
Post-condición	No hay

Operación	obtenerDatosClientes(): String Parámetros de entrada:
Descripción	Obtiene los datos que se encuentran en el archivo "cuentas.txt"
Pre-condición	-Debe de existir el archivo de texto "cuenta.txt"
Post-condición	No hay

Operación	obtenerDatosRecaudaciones(): String Parámetros de entrada:
Descripción	Obtiene la recaudaciones realizadas
Pre-condición	-Debe de existir la recaudación de un personaje
Post-condición	No hay

Link GitHub: [https://github.com/Zenthio/Taller1\\_Peralta\\_Morales](https://github.com/Zenthio/Taller1_Peralta_Morales)