

# Primer Parcial

## Ingeniería de Software II

Farizano, Juan Ignacio. Legajo: F-3562/9

---

## Índice

<b>1. Problema 1</b>	<b>2</b>
1.1. Items de cambio . . . . .	2
1.2. Especificación de interfaces . . . . .	2
1.3. Estructura de módulos . . . . .	3
1.4. Guía de módulos . . . . .	4
1.5. Estrategia de cambio para incorporar un nuevo idioma . . . . .	5
<b>2. Problema 2</b>	<b>6</b>
2.1. Especificación de interfaces . . . . .	6
2.2. Guía de módulos . . . . .	7
<b>3. Problema 3</b>	<b>8</b>
3.1. Punto a . . . . .	8
3.2. Punto b . . . . .	8
3.3. Punto c . . . . .	8

# 1. Problema 1

## 1.1. Items de cambio

- Campos de datos del formulario
- Presentación en diferentes idiomas del formulario
- Entrada de datos
- Presentación al usuario en pantalla
- Presentación, ingreso de datos y rellenado del formulario
- Consultas al repositorio de datos
- Control de ingreso formulario y chequeo de datos ingresados con el repositorio. (Proceso principal)

## 1.2. Especificación de interfaces

<b>Module</b>	<b>DatosFormulario</b>
<b>imports</b>	DNI
<b>exportsproc</b>	setDNI( <b>i</b> DNI) getDNI() : DNI setNombre( <b>i</b> String) getNombre() : String
<b>comments</b>	Considero al tipo DNI como un tipo básico.

<b>Module</b>	<b>LocalizacionForm</b>
<b>exportsproc</b>	titulo():String etiquetaNombre():String etiquetaDNI():String etiquetaBotonAceptar():String
<b>comments</b>	Para cada texto o etiqueta que se encuentre en el formulario se presenta un procedimiento.

<b>Module</b>	<b>LocalizacionForm_SP inherits from <a href="#">LocalizacionForm</a></b>
<b>comments</b>	Localización del formulario de ingreso de datos en español.

<b>Module</b>	<b>LocalizacionForm_PT inherits from <a href="#">LocalizacionForm</a></b>
<b>comments</b>	Localización del formulario de ingreso de datos en portugués.

<b>Module</b>	<b>LocalizacionForm_EN inherits from <a href="#">LocalizacionForm</a></b>
<b>comments</b>	Localización del formulario de ingreso de datos en inglés.

<b>Module</b>	<b>Pantalla</b>
<b>exportsproc</b>	dibujar()

<b>Module</b>	<b>IngresoDeDatos</b>
<b>exportsproc</b>	leerEntrada()

<b>Module</b>	<b>Formulario</b>
<b>imports</b>	<a href="#">DatosFormulario</a> , <a href="#">LocalizacionForm</a> , <a href="#">Pantalla</a> , <a href="#">IngresoDeDatos</a>
<b>exportsproc</b>	Formulario( <b>i</b> <a href="#">LocalizacionForm</a> ) mostrarFormulario() rellenarFormulario()

<b>Module</b>	<b>RepositorioDeDatos</b>
<b>imports</b>	DNI
<b>exportsproc</b>	existeDNI( <b>i</b> DNI) : Bool corresponde( <b>i</b> DNI, <b>i</b> String) : Bool

<b>Module</b>	<b>RepositorioDeDatos_BDRelacional</b> inherits from <a href="#">RepositorioDeDatos</a>
<b>comments</b>	Repositorio de datos implementado con una base de datos relacional

<b>Module</b>	<b>RepositorioDeDatos_ArchivoTexto</b> inherits from <a href="#">RepositorioDeDatos</a>
<b>comments</b>	Repositorio de datos implementado con un archivo de texto

<b>Module</b>	<b>ProcesoFormulario</b>
<b>imports</b>	<a href="#">Formulario</a> , <a href="#">RepositorioDeDatos</a>
<b>exportsproc</b>	recibirFormulario( <b>i</b> String) : <a href="#">Formulario</a> revisarFormulario( <b>i</b> <a href="#">Formulario</a> ) : Bool
<b>comments</b>	recibirFormulario() recibe un string representando el idioma en el que se presentará el texto; revisarFormulario() revisa con la base de datos que los datos ingresados corresponden.

### 1.3. Estructura de módulos

<b>Module</b>	<b>ProcesoFormulario</b>
<b>comprises</b>	<a href="#">Formulario</a> <a href="#">RepositorioDeDatos</a> <a href="#">RepositorioDeDatos_BDRelacional</a> <a href="#">RepositorioDeDatos_ArchivoTexto</a>
<b>exportsproc</b>	Formulario, RepositorioDeDatos

<b>Module comprises</b>	<b>Formulario</b> <a href="#">DatosFormulario</a> <a href="#">LocalizacionForm</a> <a href="#">LocalizacionForm_SP</a> <a href="#">LocalizacionForm_PT</a> <a href="#">LocalizacionForm_EN</a> <a href="#">Pantalla</a> <a href="#">IngresoDeDatos</a>
<b>exportsproc</b>	<a href="#">DatosFormulario</a> , <a href="#">LocalizacionForm</a> , <a href="#">Pantalla</a> , <a href="#">IngresoDeDatos</a>

## 1.4. Guía de módulos

El programa consiste de un módulo que se describe a continuación.

### 1.4.1. ProcesoFormulario

Este módulo contiene los módulos necesarios para poder presentar al usuario un formulario con un idioma especificado previamente, su relleno con los datos pedidos y posteriormente la verificación de estos datos ingresados en un repositorio de datos.

**1.4.1.1. Formulario** Este módulo provee la interfaz para que al usuario se le presente el formulario en el idioma correspondiente y este complete con sus datos.

**1.4.1.1.1. DatosFormulario** Oculta la estructura de datos utilizada para almacenar la información del usuario.

**1.4.1.1.2. LocalizacionForm** Módulo abstracto que provee una interfaz para que los textos del formulario puedan ser presentados en diferentes idiomas.

**1.4.1.1.3. LocalizacionForm\_SP** Oculta la localización del texto del formulario en el idioma español.

**1.4.1.1.4. LocalizacionForm\_PT** Oculta la localización del texto del formulario en el idioma portugués.

**1.4.1.1.5. LocalizacionForm\_EN** Oculta la localización del texto del formulario en el idioma Inglés.

**1.4.1.1.6. Pantalla** Este módulo lógico oculta la interfaz software/hardware que se utiliza para mostrar en pantalla el programa.

**1.4.1.1.7. IngresoDeDatos** Este módulo lógico oculta la interfaz software/hardware que se utiliza para que el usuario pueda ingresar sus datos.

**1.4.1.2. RepositorioDeDatos** Este módulo abstracto oculta la interfaz de software de la estructura de datos utilizada para almacenar los datos de los usuarios.

**1.4.1.3. RepositorioDeDatos\_BDRelacional** Oculta la interfaz utilizada para realizar consultas a una base de datos relacional.

**1.4.1.4. RepositorioDeDatos\_ArchivoTexto** Oculta la interfaz utilizada para realizar consultas a un archivo de texto.

## **1.5. Estrategia de cambio para incorporar un nuevo idioma**

Para incorporar un nuevo idioma al sistema se debe definir un nuevo módulo **LocalizacionForm\_IDIOMA** que herede la interfaz del módulo **LocalizacionForm**, donde cada procedimiento debe devolver un dato de tipo String que contenga la traducción del texto/etiqueta correspondiente.

## 2. Problema 2

### 2.1. Especificación de interfaces

<b>Module</b>	<b>CuentaBancaria</b>
<b>imports</b>	NumCta, Monto
<b>exportsproc</b>	CuentaBancaria( <b>i</b> NumCta) depositar( <b>i</b> Monto) extraer( <b>i</b> Monto) saldo() : Monto getNumCta() : NumCta
<b>comments</b>	NumCta es un tipo básico que representa un identificador único de cada cuenta

<b>Module</b>	<b>CajaAhorros inherits from</b> <a href="#">CuentaBancaria</a>
---------------	---

<b>Module</b>	<b>CuentaCorriente inherits from</b> <a href="#">CuentaBancaria</a>
---------------	---

<b>Module</b>	<b>Transferencias</b>
<b>imports</b>	<a href="#">CuentaBancaria</a> , Monto
<b>exportsproc</b>	transferir( <b>i</b> <a href="#">CuentaBancaria</a> , <b>i</b> <a href="#">CuentaBancaria</a> , <b>i</b> Monto)

<b>Module</b>	<b>AlmacenamientoSecundario</b>
<b>imports</b>	NumCta, Monto buscarNumCta( <b>i</b> NumCta) : Monto guardarCta( <b>i</b> NumCta, <b>i</b> Monto)

<b>Generic Module</b>	<b>Lista(X)</b>
<b>imports</b>	X
<b>exportsproc</b>	add( <b>i</b> X) head():X next():X more():Bool delete()

<b>Module</b>	<b>ListaCuentas is</b> <a href="#">Lista</a> ( <a href="#">CuentaBancaria</a> )
---------------	---

<b>Module</b>	<b>ManejoDatos</b>
<b>imports</b>	<a href="#">CuentaBancaria</a> , <a href="#">AlmacenamientoSecundario</a> , <a href="#">ListaCuentas</a> , NumCta
<b>exportsproc</b>	guardarCuentas( <b>i</b> <a href="#">ListaCuentas</a> ) buscarCuenta( <b>i</b> NumCta) : CuentaBancaria

## 2.2. Guía de módulos

### 2.2.1. Transferencias

Este módulo oculta los procedimientos realizados para transferir montos de dinero entre cuentas bancarias.

**2.2.1.1. CuentaBancaria** Este módulo lógico provee la interfaz necesaria para la gestión de cuenta bancaria.

**2.2.1.2. CajaAhorros** Oculta la implementación de las operaciones bancarias realizables en una caja de ahorros.

**2.2.1.3. CuentaCorriente** Oculta la implementación de las operaciones bancarias realizables en una cuenta corriente.

### 2.2.2. ManejoDatos

Este módulo agrupa los módulos necesarios para guardar o leer cuentas bancarias y su monto en un almacenamiento secundario.

**2.2.2.1. CuentaBancaria** Este módulo lógico provee la interfaz necesaria para la gestión de cuenta bancaria.

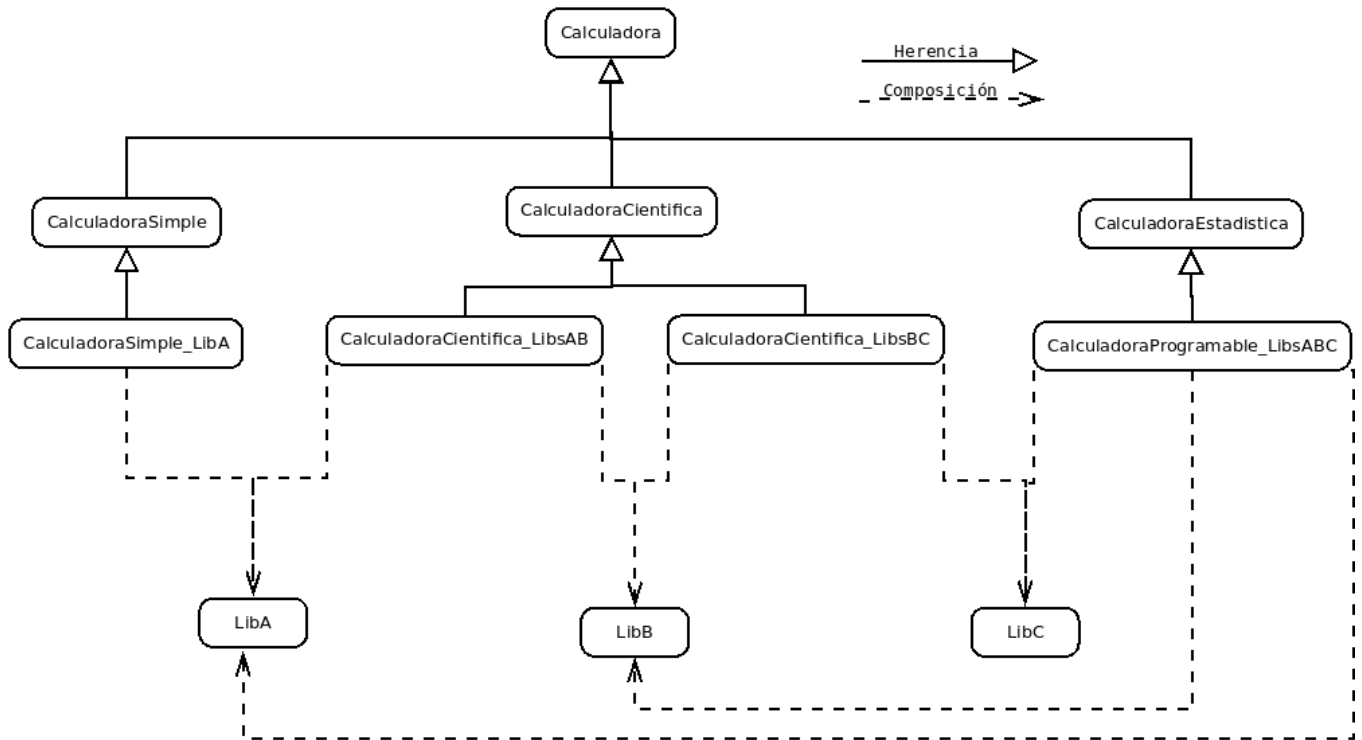
**2.2.2.2. CajaAhorros** Oculta la implementación de las operaciones bancarias realizables en una caja de ahorros.

**2.2.2.3. CuentaCorriente** Oculta la implementación de las operaciones bancarias realizables en una cuenta corriente.

**2.2.2.4. AlmacenamientoSecundario** Este módulo abstracto provee la interfaz de software/hardware necesaria para almacenar y leer datos del almacenamiento secundario

### 3. Problema 3

#### 3.1. Punto a



#### 3.2. Punto b

Si se desea implementar un nuevo tipo de calculadora se debe diseñar un nuevo módulo abstracto que herede del módulo **Calculadora** y que provea la interfaz para cada funcionalidad que ofrezca este tipo de calculadora. Para agregar una nueva biblioteca o implementar una nueva combinación de tipo de calculadora y bibliotecas, se debe diseñar un nuevo módulo físico que herede su interfaz del módulo previamente diseñado para ese tipo y se debe componer con los módulos de las bibliotecas deseadas. Por ejemplo, para una calculadora científica que utilice las librerías B y C se debe diseñar el módulo **CalculadoraCientifica\_LibsBC** que herede su interfaz del módulo **CalculadoraCientifica** y se compone con los módulos **LibB** y **LibC**.

#### 3.3. Punto c

Primero diseño un módulo abstracto que será un esqueleto de funciones comunes a todas las calculadoras como encender, apagar, mostrar los resultados, recibir entradas, etc. En este caso para simplificar solo presento los procedimientos de encender y apagar.

Module	Calculadora
exportsproc	encender() apagar()

Luego, para cada tipo de calculadora diferente diseño un módulo lógico con la interfaz que provee



las funciones correspondiente a cada tipo que se necesite implementar.

<b>Module</b> <b>exportsproc</b>	<b>CalculadoraSimple inherits from <a href="#">Calculadora</a></b> sumar( <b>i</b> Int, <b>i</b> Int) : Int restar( <b>i</b> Int, <b>i</b> Int) : Int dividir( <b>i</b> Int, <b>i</b> Int) : Int mutiplicar( <b>i</b> Int, <b>i</b> Int) : Int potenciaDos( <b>i</b> Int) : Int
-------------------------------------	--

<b>Module</b> <b>exportsproc</b>	<b>CalculadoraCientifica inherits from <a href="#">Calculadora</a></b> sumar( <b>i</b> Int, <b>i</b> Int) : Int restar( <b>i</b> Int, <b>i</b> Int) : Int dividir( <b>i</b> Int, <b>i</b> Int) : Int mutiplicar( <b>i</b> Int, <b>i</b> Int) : Int exponenciar( <b>i</b> Int, <b>i</b> Int) : Int raizCuadrada( <b>i</b> Int, <b>i</b> Int) : Int
-------------------------------------	---

<b>Generic Module</b> <b>imports</b> <b>exportsproc</b>	<b>Lista(X)</b> X add( <b>i</b> X) head():X next():X more():Bool delete()
---	---

<b>Module</b>	<b>ListaNumeros is <a href="#">Lista(Int)</a></b>
---------------	---

<b>Module</b> <b>imports</b> <b>exportsproc</b>	<b>CalculadoraEstadistica inherits from <a href="#">Calculadora</a></b> ListaNumeros sumar( <b>i</b> Int, <b>i</b> Int) : Int restar( <b>i</b> Int, <b>i</b> Int) : Int dividir( <b>i</b> Int, <b>i</b> Int) : Int mutiplicar( <b>i</b> Int, <b>i</b> Int) : Int promedio( <b>i</b> <a href="#">ListaNumeros</a> ) : Int moda( <b>i</b> <a href="#">ListaNumeros</a> ) : Int
---	---

Por último, para cada combinación de tipo de calculadoras y bibliotecas de funciones matemáticas defino un módulo que herede la interfaz del tipo de calculadora correspondiente y se compone con los módulos de las bibliotecas que se quieran utilizar

<b>Module</b> <b>imports</b>	<b>CalculadoraSimple_LibA inherits from <a href="#">CalculadoraSimple</a></b> LibA
---------------------------------	---

<b>Module</b>	<b>CalculadoraCientifica_LibAB inherits from <a href="#">CalculadoraCientifica</a></b>
<b>imports</b>	LibA, LibB

<b>Module</b>	<b>CalculadoraCientifica_LibBC inherits from <a href="#">CalculadoraCientifica</a></b>
<b>imports</b>	LibB, LibC

<b>Module</b>	<b>CalculadoraEstadistica_LibABC inherits from <a href="#">CalculadoraEstadistica</a></b>
<b>imports</b>	LibA, LibB, LibC