

## **SISTEMA DE CONTROL PIB MUNDIAL**

### **Integrantes:**

- Jhoan David Fiat Restrepo – A00359132
- Jhon Stiven Arboleda Gallego - A00359103

### **Enunciado**

Una comisión internacional nos ha pedido realizar un software para calcular el PIB de todos los países que se deseen registrar a esta comisión, simulando la mayoría de las actividades que se realizan dentro de cada país normalmente. El PIB es una magnitud de estudio dentro de la macroeconomía, la cual expresa el valor monetario de la producción de bienes y servicios dentro del país en un período determinado. Pero normalmente se calcula una vez al año, o trimestralmente. La idea que plantea esta comisión es que con el nuevo programa que se desea realizar, el cálculo del PIB se puede obtener en cualquier momento del año, cada vez que se actualice la base de datos. Por ejemplo, al registrar un cliente dentro de una empresa, al contratar empleados, fundar empresas, etc.

A cada país se le pide para registrarse ante la comisión los siguientes datos:

- Nombre.
- Cantidad de habitantes.
- Extensión territorial.
- Su presidente actual.
- El gasto público.
- Si tiene mar o no.

Cada país a su vez permite dentro de sus bases de datos registrar empresas dentro de su propio sistema, cada empresa tiene:

- Nombre.
- Identificación.
- Fecha de fundación.
- Gasto fijo.

Al gasto fijo de cada empresa se le suma el gasto de los empleados, generando un gasto total. Cada una de las empresas tiene exportaciones e importaciones, se ha definido que las exportaciones corresponden a un valor aleatorio entre el 12% y el 21% del gasto público, las importaciones corresponden al 34% por ciento de este gasto público. Para generalizar importaciones con exportaciones, se resta las importaciones a las exportaciones, generando un solo valor. Este valor se multiplica por el número de empresas para dar el valor más general de las exportaciones e importaciones que han ocurrido en cada país. Este valor se suma con el gasto público, con el gasto de cada cliente y con el gasto final de cada una de las empresas del país. Calculando el PIB.

Cada empresa tiene a sus respectivos empleados, al contratar un empleado se le pide:

- Nombre.
- Identificación.
- Su salario.
- Cargo.

Deben estar ordenados por su número de identificación, el salario de los empleados se le suma al gasto fijo de la empresa, obteniendo el gasto total.

Las empresas pueden ser con animo de lucro o sin animo de lucro. Y a su vez las empresas sin animo de lucro pueden ser empresas de educación; de las cuales se conoce su tipo de educación, si es colegio, universidad, instituto, etc. Y el nombre de su rector. Otro tipo de empresa sin animo de lucro es la cooperativa, de la cual se conoce su objetivo y su calificación. Y el último tipo de empresa sin animo de lucro es la fundación de la cual sabemos la ayuda que recibe en dólares a su causa. Únicamente las empresas de educación y las cooperativas pagan impuestos. Las de educación pagan un impuesto del 19% de su gasto total. Las cooperativas pagan un impuesto según su calificación, la cual es un número entero entre 1 y 10. En donde la mejor calificación es el 1 y la peor el 10. Este valor se multiplica por el gasto total de cada empresa y se divide por 100.

Las empresas con animo del lucro tienen una ganancia siempre. Todas pagan un impuesto del 19%, las empresas con animo de lucro pueden ser:

- Financieras.
- Cadenas de supermercados.
- Empresas de transporte.

De las empresas financieras conocemos la tasa de interés que manejan y si es totalmente legal o no. Por ejemplo, un gota a gota no se considera legal, sin embargo, debe ser tomado en cuenta en los movimientos económicos del país. Si esta empresa es legal, debe estar regulada por entidades nacionales y no puede tener una tasa de interés mayor al 13%. Pero las empresas que trabajan en la ilegalidad no tienen quien las regule y manejan el interés que deseen.

De las cadenas de supermercados conocemos el nombre de su dueño, su calificación y la cantidad de supermercados que existen a nivel nacional. Las empresas de transporte tienen sus vehículos, de los cual se conoce:

- Su marca.
- Identificación.

Cada vehículo puede ser terrestre, marítimo o aéreo.

Para calcular el gasto final de una empresa, se suma su impuesto junto con su gasto total. Por esto mismo en una fundación el gasto final es igual al gasto total. Y de la misma forma ocurriría con una cooperativa con calificación 1.

Cada empresa tiene registrados en su base de datos a todos sus clientes, que en la comisión serán todos tratados como clientes. De los cuales conocemos:

- Su nombre.

- Identificación.
- Gasto.

Cada cliente puede ser adulto, niño o mascota. De los adultos conocemos su gasto en licores, ya que se prohíbe el expendio de bebidas embriagantes a menores de edad. Este valor se le suma al gasto de cada cliente. De los niños solo podemos conocer la información de su estudio, ya que ellos solamente deben concentrarse en estudiar. Se conoce si el niño recibe educación privada o pública y su gasto en ella. Si el niño recibe educación privada el gasto en dicha educación se le suma al gasto del cliente. De las mascotas se conoce si tiene pedigree o no, el gasto de la mascota corresponde a 500 dólares. Sin embargo, si una mascota tiene el certificado de pedigree, su gasto corresponde a 1000 dólares.

No puede existir ningún objeto de la misma clase con un ID igual, de resto todo puede ser igual, y sea cual sea como se guardan, siempre estarán ordenados por su ID. Exceptuando a los países, que pueden estar ordenados por cualquier criterio que se desee.

## Requerimientos Funcionales

1. Registrar los datos de los ciudadanos de cada país con sus debidos datos, como lo son el nombre, la identificación, país y sus gatos, los cuales dependerá que dependerá de si es un adulto, niño o mascota, ya que si es un adulto tendrá gastos en licor, si es niño tendrá gasto en educación y si es mascota deberá agregar su pedigree.
  - 1.1. **RNF:** La clase ciudadano es clase padre de mascota, niño y adulto.
2. Registrar un país con todos sus datos correspondientes, el nombre, cantidad de habitantes, extensión territorial, nombre del presidente, los gatos públicos y en caso de tener, su mar principal.
  - 2.2 **RNF:** Los países están organizados en un ArrayList dentro del software.
3. Registrar una empresa con sus datos como lo son el nombre, identificación, fecha de creación, gastos y su tipo, de acuerdo a este tendrá unas características diferentes. Si es con ánimo de lucro: ganancias. Si es financiera, tasa de interés, permiso. Si es una cadena de supermercados, dueño, calificación, cantidad de supermercados. Si es de educación, tipo de institución, rector. Si es una cooperativa, objetivo, calificación. Si es una fundación, ayuda, país
  - 3.1. **RNF:** La clase empresa es clase padre de las clases empresas con ánimo de lucro y de empresas sin ánimo de lucro.
  - 3.2. **RNF:** La clase empresa con ánimo de lucro es clase padre de las clases cadena de supermercados, financiera y de transporte.
  - 3.3. **RNF:** La clase empresa sin ánimo de lucro es clase padre de las clases educación, cooperativa y fundación.
4. Registrar un empleado con sus datos correspondientes, como su nombre, su identificación, su cargo y su salario.

**4.1. RNF:** Los empleados están organizados en la empresa con una lista doblemente enlazada.

**5.** Registrar un vehículo junto con su marca, su tipo y su identificación.

**5.1 RNF:** Las empresas de transporte tienen un árbol binario de búsqueda de vehículos.

**6.** Registrar el gasto total de la empresa, sumando el gasto de la empresa con el salario de todos los empleados.

**7.** Calcular las importaciones, las cuales corresponden al 34% del gasto de la empresa.

**7.1. RNF:** El cálculo de las importaciones se realiza bajo una interfaz llamada Imports. Bajo el método `calculateImports()`.

**8.** Calcular las exportaciones, las cuales corresponden a un porcentaje aleatorio entre el 1% y 21% del gasto que tiene la empresa.

**8.1. RNF:** El cálculo de las exportaciones se realiza bajo una interfaz llamada Exports. Bajo el método `calculateExports()`.

**9.** Generalizar las importaciones y exportaciones, para esto se le debe restar a las exportaciones, las importaciones.

**10.** Calcular el PIB, el cual se calcula sumando los gastos públicos, el gasto total de cada ciudadano y el gasto final de cada empresa, así se obtiene el PIB del país.

**11.** Calcular el gasto final de la empresa, este cálculo depende si la empresa es con ánimo de lucro: se calcula gasto total más las ganancias más la generalización de las importaciones y exportaciones. Si es sin ánimo de lucro: gasto total más los impuestos a pagar más la generalización de las exportaciones e importaciones. Si es fundación: Gasto menos ayuda más la generalización de exportaciones e importaciones.

**12.** Calcular los impuestos de una empresa de educación el cual corresponde al 7% del gasto de la empresa

**13.** Calcular los impuestos de una cooperativa el cual corresponde a la calificación de la cooperativa (que es un valor entero entre el 1 y el 10) por el gasto total de cada empresa y se divide entre 100.

**14.** Calcular el gasto del adulto el cual corresponde al gasto más el gasto en licor.

**15.** Calcular el gasto niño es el gasto más el gasto en educación. Solo si este tiene educación privada, de lo contrario solo sería el gasto.

16. Calcular el gasto de mascota, será el gasto fijo de la mascota por dos y se le suma el gasto, esto en caso de tener pedigree, en caso de no tener será el gasto más el gasto fijo.
17. Imprimir el nombre de los países se imprimirá en orden. El criterio por el cual serán ordenados los elige el usuario.

## RNF

1. EL gasto total de la clase ciudadano se realiza bajo la interfaz Spending con el método calculateTotalSpending().
2. Todos los árboles binarios de búsqueda estarán ordenados por la identificación del objeto. Las listas enlazadas no estarán ordenadas bajo ningún criterio. Solamente por el orden en el que se vayan registrando. Ya que los objetos de las listas se van agregando al final de la lista.
3. No pueden existir dos objetos de la misma clase con el mismo numero de identificación, se arroja una excepción.
4. Los ciudadanos están organizados en forma de árbol binario de búsqueda dentro del país.
5. Las empresas están organizadas en forma de lista simplemente enlazada dentro del país.
6. Nunca ha ocurrido que dos presidentes de dos países diferentes tengan el mismo nombre, por lo cual no está permitido registrar un país si el nombre de su presidente ya se encuentra en el sistema.
7. Para las agencias internacionales, solo puede existir un nombre por país, por lo cual si se desea agregar un país con un nombre que ya está en el programa. Una excepción de lanzará.

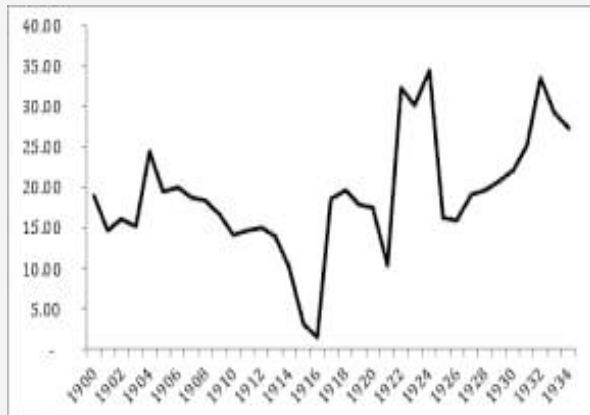
## DISEÑO INTERFAZ PROGRAMA- MOCKUPS.





# DAR PIB

▼ LISTADO DE PAÍSES
COLOMBIA
MEXICO
COSTA RICA
EE.UU
ESPAÑA
INGLATERRA



# ORDENAR PAÍSES

► CRITERIOS

► LISTADO DE PAÍSES



### Justificación

Es un programa lo suficientemente complejo para dos integrantes, con una cantidad considerable de excepciones. Un excelente manejo de herencias y desacoplamiento a través de interfaces. Además, consideramos que el proyecto podría tener un impacto muy importante si se trabaja a futuro, ya que el PIB es un indicador económico vital para las personas que trabajan en la rama de la economía nacional y mundial. Y nuestro programa llevado a un nivel mucho más avanzado podría facilitar muchos factores económicos. Tiene un adecuado número de restricciones para manejarse dentro del programa, suficientes para poder simular el comportamiento interno de un país de la manera mas acertada posible.



## DISEÑO DE CASOS DE PRUEBA

### Configuración de los Escenarios.

Nombre	Clase	Escenario
setupSoftwareStage	SoftwareTest	<p>Un objeto de la clase Software con 3 objetos de la clase Country:</p> <ul style="list-style-type: none"><li>- El primero con name="Colombia", population=123, extension=456.78, president="Uribe", publicSpending= 56, sea="Pacífico</li><li>- El segundo con name=" Venezuela", population=23, extension= 4556.78, president="Maduro", publicSpending= 428, sea="Pacífico</li><li>- El tercero con name="Japón", population=56123, extension= 4456.78, president="Dan-Chi", publicSpending= 6556, sea=" Indico</li></ul>

Nombre	Clase	Escenario
setupStage	CountryTest	Un objeto de la clase Country con name=" Perú", population=3234555, extension= 2463563, president=" Fiat", publicSpending=683466, sea="Pacífico

## Diseño de Casos de Prueba.

<b>Objetivo de la Prueba: (testSortCountriesByName)</b> Verificar que el método sortCountriesByName de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio de su nombre. El método se encarga de ordenar estos países por orden alfabético dentro de la colección.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesByName	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Colombia. 2- Japón. 3- Venezuela.

<b>Objetivo de la Prueba: (testSortCountriesByPopulation)</b> Verificar que el método sortCountriesByPopulation de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio de su población. El método se encarga de ordenar estos países de menor a mayor. Del que tiene menos habitantes a más habitantes.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesByPopulation	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Colombia. 2- Venezuela. 3- Japón.

**Objetivo de la Prueba: (testSortCountriesByExtension)** Verificar que el método sortCountriesByExtension de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio de su extensión territorial. El método se encarga de ordenar estos países de menor a mayor. Del que tiene menos extensión a mayor extensión.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesByExtension	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Colombia. 2- Japón. 3- Venezuela.

**Objetivo de la Prueba: (testSortCountriesByPresident)** Verificar que el método sortCountriesByPresident de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio del nombre de su presidente. El método se encarga de ordenar estos países por orden alfabético dentro de la colección.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesByPresident	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Japón. 2- Venezuela. 3- Colombia.

**Objetivo de la Prueba: (testSortCountriesByPublicSpending)** Verificar que el método sortCountriesByPublicSpending de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio de su gasto público. El método se encarga de ordenar estos países de menor a mayor. Del que tiene menos gasto a mayor gasto.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesByPublicSpending	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Venezuela 2- Colombia. 3- Japón.

**Objetivo de la Prueba: (testSortCountriesBySea)** Verificar que el método sortCountriesBySea de la clase Software funcione correctamente, al intentar ordenar los objetos agregados a la colección llamada countries por el criterio del nombre de su océano principal. El método se encarga de ordenar estos países por orden alfabético dentro de la colección.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	sortCountriesBySea	setupSoftwareStage	Ninguno	Se ordenan los objetos quedando en el orden:  1- Japón. 2- Colombia. 3- Venezuela.

**Objetivo de la Prueba: (testBinarySearchCountryByName)** Verificar que el método binarySearchCountryByName de la clase Software funcione correctamente, al intentar buscar un elemento dentro de la colección de países dando su determinado nombre, ya que en la colección no hay países que se llamen igual. Este método retorna un objeto de tipo Country.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	binarySearchCountryByName	setupSoftwareStage	Colombia	Retorna un objeto con name="Colombia", population=123, extension=456.78, president="Uribe", publicSpending= 56, sea="Pacífico

**Objetivo de la Prueba: (testBinarySearchCountryByPresident)** Verificar que el método binarySearchCountryByPresident de la clase Software funcione correctamente, al intentar buscar un elemento dentro de la colección de países dando su determinado presidente, ya que en la colección no hay países que tengan un presidente con el mismo nombre. Este método retorna un objeto de tipo Country.

Clase	Método	Escenario	Valores de Entrada	Resultado
Software	binarySearchCountryByPresident	setupSoftwareStage	Maduro	Retorna un objeto con name="Venezuela", population=23, extension= 4556.78, president="Maduro", publicSpending= 428, sea="Pacífico

**Objetivo de la Prueba: (addEnterpriseTest)** Verificar que el método addEnterprise de la clase Country funcione correctamente, al intentar agregar los objetos agregados a su determinada estructura de dator. El método se encarga de agregar estos objetos. Y de lanzar una excepción si algún objeto con la misma ID ya ha sido agregado previamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Country	addEnterprise	setupStage	<p>Tres objetos de la clase Enterprise:</p> <ul style="list-style-type: none"> <li>- De tipo Transport → name="Confenalco", ID= 535336, creationDate="22/11/1980", spending=65435.76, earnings= 54678</li> <li>- De tipo Education → name="Icesi", ID= 5675, creationDate="22/05/1990", spending=56993, type="University", rector= "Piedrahita"</li> <li>- De tipo Financial → name="Davivienda", ID= 123, creationDate= "12/07/1999", spending=456675, earnings= 5555, interestRate=12, permission= false</li> </ul>	Todos los objetos son agregados mientras no se repitan.

**Objetivo de la Prueba: (addCitizenTest)** Verificar que el método addCitizen de la clase Country funcione correctamente, al intentar agregar los objetos agregados a su determinada estructura de dator. El método se encarga de agregar estos objetos. Y de lanzar una excepción si algún objeto con la misma ID ya ha sido agregado previamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Country	addCitizen	setupStage	<p>Tres objetos de la clase Citizen:</p> <ul style="list-style-type: none"> <li>- De tipo Pet → name=" Pepe", ID= 3456, spending =5678, pedigree= true</li> <li>- De tipo Adult→ name=" Daniel", ID= 64643456, spending =5678</li> <li>- De tipo Child→ name=" Juan", ID= 1002567502, spending =34567,</li> </ul>	Todos los objetos son agregados mientras no se repitan.

			privateEducation= true, educationExpense= 345	
--	--	--	--	--

**Objetivo de la Prueba: (searchCitizenTest)** Verificar que el método searchCitizen de la clase Country funcione correctamente, al intentar buscar un objeto de la clase Citizen ordenado en una estructura dentro de la clase Country. Dado un ID se retorna un objeto. Si el objeto no existe se retorna null.

Clase	Método	Escenario	Valores de Entrada	Resultado
Country	searchCitizen	setupStage	<ul style="list-style-type: none"> <li>- "1"</li> <li>- "3456"</li> <li>- "64643456"</li> </ul>	<ul style="list-style-type: none"> <li>- NULL</li> <li>- Un objeto de la clase Citizen de tipo Pet → name=" Pepe", ID= 3456, spending =5678, pedigree= true</li> <li>- Un objeto de la clase Citizen de tipo Adult → name=" Daniel", ID= 64643456, spending =5678</li> </ul>