

ESCUELA COLOMBIANA DE INGENIERÍA DESARROLLO ORIENTADO POR OBJETOS

Diseño y Pruebas. Interacción entre objetos.

2026-1 — Laboratorio 2/6

OBJETIVOS

Desarrollar competencias básicas para:

1. Desarrollar una aplicación aplicando Desarrollo Orientado al Comportamiento (BDD) y Desarrollo Dirigido por Modelos (MDD).
2. Realizar diseños (ingeniería directa e inversa) utilizando una herramienta de modelado (astah)
3. Manejar pruebas de unidad usando un framework ([junit](#))
4. Apropiar nuevas clases consultando sus especificaciones ([API java](#))
5. Experimentar las prácticas XP :
Designing. Use [CRC](#) cards for design sessions.
Testing. All code must have [unit tests](#).

ENTREGA

- Incluyan en un archivo **.zip** los archivos correspondientes al laboratorio. El nombre debe ser los apellidos de los dos miembros del equipo ordenados alfabéticamente y separados por un guion. (Casallas-Duarte)
- Publiquen el avance al final de la sesión y la versión definitiva en la fecha indicada, en los espacios correspondientes.

CONTEXTO

Objetivo: Desarrollar una versión simplificada de [Pandas](#): [BabyPandas](#)

[Pandas](#) es una biblioteca de [Python](#) para manipulación y análisis de datos, basada en estructuras como [Series](#) y [DataFrames](#) que permiten trabajar con datos tabulares de forma eficiente. En este laboratorio desarrollaremos una versión simplificada de [Pandas](#) con únicamente [DataFrames](#). Un [DataFrame](#) es una estructura de datos de dos dimensiones (rectangulares) que pueden contener datos de diferentes tipos, por lo tanto, son heterogéneas.
[Pandas Tutorial](#)

Column names			
	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

1. Conociendo el proyecto

A Revisando línea base

[En lab02.doc]

1. El proyecto “[babyPandas](#)” contiene una construcción parcial del sistema. Revisen el directorio donde se encuentra el proyecto. Describan el contenido en términos de (a) directorios y de (b) las extensiones de los archivos.
2. Explore el proyecto en [BlueJ](#) (a) ¿Cuántas clases tiene? (b) ¿Cuál es la relación entre ellas? (c) ¿Cuál es la clase principal de la aplicación? (d) ¿Cómo la reconocen? (e) ¿Cuál es la clase “diferente”? (f) ¿Cuál es su propósito?
Para las siguientes dos preguntas sólo consideren las clases “normales”:
3. Generen y revisen la documentación del proyecto: ¿está completa la documentación de cada clase? (Detallen el estado de documentación: encabezado y métodos)
4. Revisen el código fuente del proyecto, ¿en qué estado está cada clase? (Detallen el estado de las fuentes considerando dos dimensiones: (a) la primera, atributos y métodos, y la segunda, (b) código, documentación y comentarios) (c) ¿Qué diferencia hay entre el código, la documentación y los comentarios?

B Realizando ingeniería inversa

[En lab02.doc babyPandas .astah]

MDD: MODEL DRIVEN DEVELOPMENT

1. Completen el diagrama de clases correspondiente al proyecto. (No incluyan la clase de pruebas)
2. (a) ¿Cuáles contenedores están definidos? (b) ¿Qué diferencias hay entre el nuevo contenedor y el ArrayList y el vector [] que conocemos? Consulte el API de java.
3. En el nuevo contenedor, (a) ¿Cómo adicionamos un elemento? (b) ¿Cómo lo consultamos? (c) ¿Cómo lo eliminamos?

2. Aprendiendo a probar

A Conociendo Pruebas en BlueJ

[En lab02.doc *.java]

BDD: BEHAVIOUR DRIVEN DEVELOPMENT

Para poder cumplir con la prácticas XP vamos a aprender a realizar las pruebas de unidad usando las herramientas apropiadas. Para eso implementaremos algunos métodos en la clase `DataFrame`.

1. Revisen el código de la clase `DataFrameTest` (a) ¿cuáles etiquetas tiene (componentes con símbolo @)? (b) ¿cuántos métodos tiene? (c) ¿cuántos métodos son de prueba? (d) ¿cómo los reconocen?
2. Ejecuten los tests de la clase `DataFrameTest`. (click derecho sobre la clase, Test All) (a) ¿cuántas pruebas se ejecutan? (b) ¿cuántas pasan? (c) ¿por qué? (d) Capturen la pantalla.
3. Estudien las etiquetas encontradas en 1. Expliquen en sus palabras su significado.
4. Estudie los métodos `assertTrue`, `assertFalse`, `assertEquals`, `assertNull` y `fail` de la clase `Assert` del API JUnit 1. Explique en sus palabras que hace cada uno de ellos.
5. Investiguen y expliquen la diferencia que entre un fallo y un error en `JUnit`. Escriba código, usando los métodos del punto 4., para codificar los siguientes tres casos de prueba y lograr que se comporten como lo prometen `shouldPass`, `shouldFail`, `shouldErr`.

B Practicando Pruebas en BlueJ

[En lab02.doc, `DataFrame`, *.java]

BDD: BEHAVIOUR DRIVEN DEVELOPMENT

Ahora vamos escribir el código necesario para que las pruebas de pasen `DataFrameTest`.

1. (a) Determinen los atributos de la clase `DataFrame`. (b) Justifiquen la selección.
2. (a) Determinen el invariante de la clase `DataFrame`. (b) Justifiquen la decisión.
3. Implementen los métodos de `DataFrame` necesarios para pasar todas las pruebas definidas. (a) ¿Cuáles métodos implementaron?
4. Capturen los resultados de las pruebas de unidad.

3. Desarrollando el proyecto babyPandas.

[En lab02.doc, `desi`, *.java] BDD - MDD Para desarrollar esta aplicación vamos a considerar algunos ciclos.

En cada ciclo deben realizar los pasos definidos a continuación.

- I Definir los métodos base de correspondientes al mini-ciclo actual.
- II Definir y programar los casos de prueba de esos métodos.
Piensen en los debería y los noDebería (`should...` y `shouldNot...`)
- III Diseñar los métodos
Usen diagramas de secuencia. En astah, creen el diagrama sobre el método correspondiente.
- IV Escribir el código correspondiente (no olvide la documentación)
- V Ejecutar las pruebas de unidad (vuelva a 3 (a veces a 2), si no están en verde)
- VI Completar la tabla de clases y métodos. (Al final del documento)

Ciclo 1 : Operaciones básicas: definir una variable, asignar a una variable un `DataFrame`, consultar el tamaño, consultar las primeras filas(`head`)

Ciclo 2: Operaciones unarias: seleccionar filas y columnas, seleccionar las filas que cumplen una condición

Ciclo 3 : Operaciones binarias: concatenar por filas y columnas

BONO Ciclo 4 : Defina e implemente tres nuevas operaciones

Completen la siguiente tabla indicando el número de ciclo y los métodos asociados de cada clase.

Ciclo	BabyPandas	BabyPandasTest
-------	------------	----------------

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas)
2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?
3. Considerando las prácticas XP del laboratorio, ¿cuál fue la más útil? ¿por qué?
4. ¿Cuál consideran fue el mayor logro? ¿Por qué?
5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?
6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?
7. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.