

# Xóchitl Analí Cabañas Mota

NAO ID: 3319

September 24th, 2025

In-Mexico Program Backend  
Developer Certification

Back End in Java for Information  
Processing:

**GitHub repository link**

GitHub repository link:

<https://github.com/anacasx/converter>

The screenshot shows the GitHub repository page for 'converter' by user 'anacasx'. The repository is public and has 1 commit. The main content area displays a list of files and folders: .idea, sample\_data, src, \_plugins, README.md, and pom.xml. The README.md file is selected and its content is displayed below. The README describes a Java 17 project called 'Converter' that transforms JSON files into CSV format. It includes a list of features: Java classes for reading JSON and writing CSV files, usage of Jackson for JSON parsing and OpenCSV for CSV generation, unit tests implemented with JUnit 5, and sample input and output files located in the 'sample\_data' folder. The project demonstrates error handling, isolated testing, and Javadoc documentation for clarity and maintainability. The right sidebar contains sections for 'About', 'Releases', 'Packages', 'Languages', and 'Suggested workflows'.

**converter** Public

anacasx · Delete sample\_data/output.csv · 1 commit · 1 commit · 1 commit

main · 1 branch · 1 day

Go to file · Add file · Code

**Files**

File	Commit	Time
.idea	Initial commit - converter	12 minutes ago
sample_data	Delete sample_data/output.csv	2 minutes ago
src	Initial commit - converter	12 minutes ago
_plugins	Initial commit - converter	12 minutes ago
README.md	Create README.md	2 minutes ago
pom.xml	Initial commit - converter	12 minutes ago

**README**

## JSON to CSV Converter

This repository contains a Java 17 project called Converter, developed to transform JSON files into CSV format. It includes:

- Java classes for reading JSON and writing CSV files.
- Usage of Jackson for JSON parsing and OpenCSV for CSV generation.
- Unit tests implemented with JUnit 5.
- Sample input and output files located in the `sample_data` folder.

The project demonstrates error handling, isolated testing, and Javadoc documentation for clarity and maintainability.

**About**

No description, website, or topics provided.

**Releases**

No releases published.

**Packages**

No packages published.

**Languages**

Java 100.0%

**Suggested workflows**

Build and test a Java project with Maven

**Java with Maven** Configure