

IMPLEMENTACIÓN DE UN PIPELINE CI/CD CON DOCKER Y JENKINS

PARA OPTIMIZAR UN SISTEMA DE GESTIÓN DE MOROSIDAD

Alexander Huarancca, María Mendoza, Fiorella Tamariz

01



NORTE - ANDINO
SERVICIO DE TRANSPORTE URBANO

CASO DE ESTUDIO

Y PROBLEMA DE NEGOCIO

La empresa Norte Andino SAC, con sede en Huaraz, opera en el rubro de transporte de taxis y gestiona líneas a través de tres tipos de clientes: socios, concesionarios y alquileres en línea.

Actualmente Norte Andino SAC enfrenta:

- Limitaciones tecnológicas que no siguen el ritmo de crecimiento del negocio.
- Alta dependencia de aplicación móvil combinada con múltiples archivos de Excel que fragmenta la información.
- Creciente volumen de registros y transacciones.

CONCEPTOS IMPORTANTES

01

Procesamiento secuencial y
computación paralela

02

OpenMP

03

Single Instruction Multiple
Data (SIMD)

04

Speedup

05

Integración continua (CI) y
entrega continua (CD)

06

Pipeline

APLICACIÓN

DE COMPUTACIÓN DE ALTO DESEMPEÑO

Se utilizara la paralelización para poder aprovechar la capacidad computacional y poder realizar los cambios en el estado de morosidad en registros con los que cuenta Norte Andino SAC.

Para optimizar el proceso de gestión de morosidad, se implementar a un pipeline CI/CD utilizando Docker y Jenkins.

03

OBJETIVO GENERAL

Desarrollar e implementar un pipeline CI/CD con Docker y Jenkins para optimizar y automatizar el proceso de cálculo y actualización del estado de morosidad de los clientes de Norte Andino SAC.

04

05


01

Implementar un sistema de integración continua (CI) utilizando Docker y Jenkins para automatizar las pruebas, la compilación y el despliegue del módulo de cálculo de morosidad.


03

Asegurar la portabilidad de la contenedorización completa en Docker para garantizar su despliegue en la nube.


02

Optimizar el cálculo y actualización del estado de morosidad mediante el uso de computación paralela (OpenMP), demostrando una reducción aproximada del 60% en el tiempo de procesamiento.

OBJETIVOS ESPECÍFICOS

A

Accelerated Financial Algorithms:
Derivative Pricing and Risk
Management Applications.

B

Parallel Computing in Economics
– An Overview of the Software
Frameworks.

C

Parallel computing in finance for
estimating risk-neutral densities
through option prices.

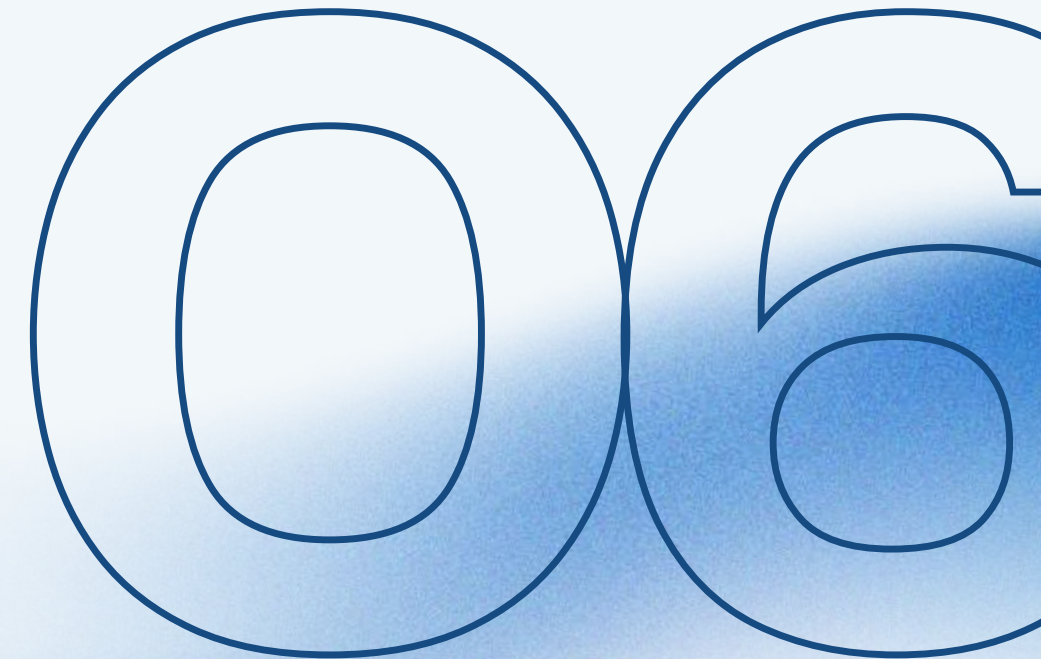
D

Practical Parallelization of
Scientific Applications with
OpenMP, OpenACC, and MPI.

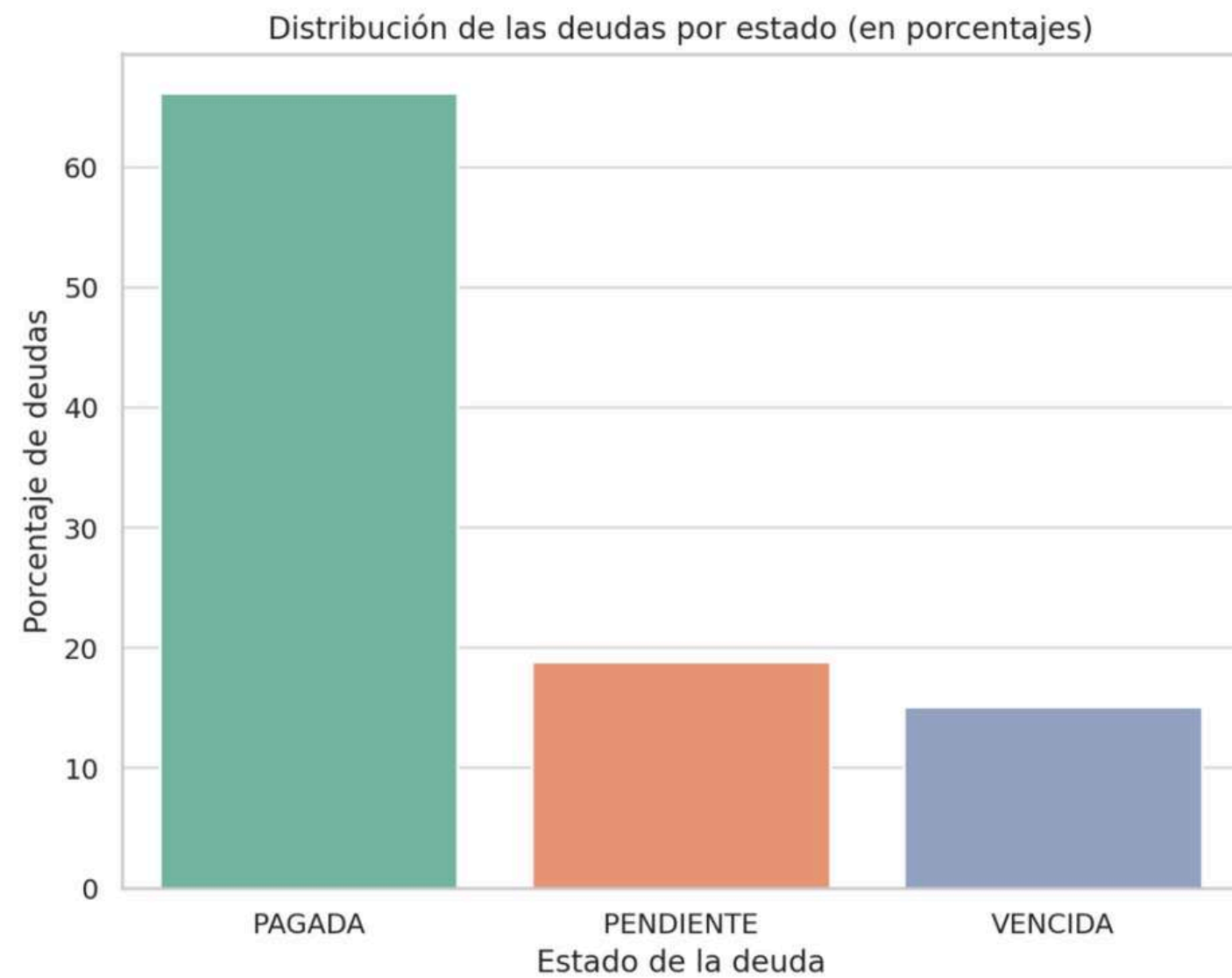
E

Automated Credit Card Default
Prediction Using MLOps on AWS:
From Pipeline Development to
Real-Time Deployment.

TRABAJOS RELACIONADOS

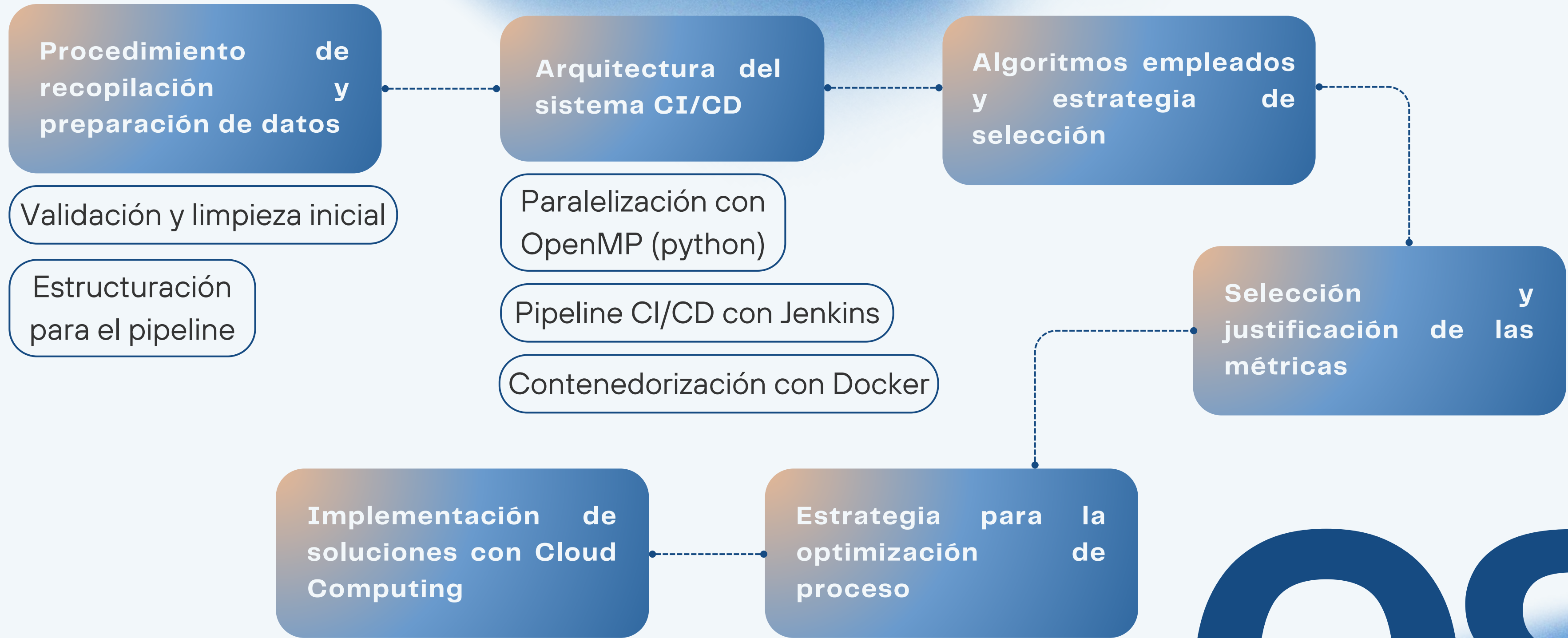


DISEÑO DEL EXPERIMENTO



DISTRIBUCION
DE LAS DEUDAS
POR ESTADO (EN
PORCENTAJES)

07



METODOLOGÍA

08

PIPELINE CI/CD CON JENKINS

- 1) CHECKOUT DEL CODIGO DESDE GITHUB
- 2) INSTALACIÓN DE DEPENDENCIAS
- 3) EJECUCIÓN DEL BENCHMARK (VALIDA LA OPTIMIZACIÓN)
- 4) ENTRENAMIENTO DEL MODELO PARALELIZADO
- 5) TESTS AUTOMATIZADOS
- 6) CONSTRUCCION DE LA IMAGEN DOCKER
- 7) PRUEBAS DEL CONTENEDOR
- 8) DESPLIEGUE AUTOMÁTICO

CONTENEDOR- RIZACIÓN CON DOCKER

DOCKER EMPAQUETA LA
APLICACIÓN COMPLETA (MODELO
ENTRENADO + API + DEPENDENCIAS)
EN UNA IMAGEN


- DESPLIEGUE
CONSISTENTE EN
DIFERENTES
SERVIDORES
- MIGRACION
FUTURA A LA NUBE
(AWS/GCP/AZURE)

- ESCALAMIENTO
HORIZONTAL
(MÚLTIPLES
CONTENEDORES)
- AISLAMIENTO DE
DEPENDENCIAS





09



Todas las pruebas de rendimiento (secuencial y paralelizado) se ejecutaron en un entorno de memoria compartida para validar la aplicabilidad de la paralelización multihilo con OpenMP.

Hardware (CPU): AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx (8 CPUs), 2.1GHz

Número de Cores físicos/lógicos: 4

Sistema Operativo: Windows 11

EXPERIMENTACIÓN Y RESULTADOS

SETUP EXPERIMENTAL

RESULTADOS DEL BENCHMARK DE PARALELIZACIÓN

09

Configuración del experimento:

Dataset: 12,512 registros, 35 features

Algoritmo: Random Forest (200 estimators)

Ejecución secuencial: n_jobs=1

Ejecucion paralela: n_jobs=-1 (todos los cores)

Tiempo Secuencia	44.53s
Tiempo Pararelización	14.98s
Speedup	2.97
Reducción	66.36%

Métrica	Valor
accuracy	0.8509744182381678
n_estimators	100
max_depth	10

Se colocó la pararelización dentro de los hiperparámetros de RandomForest, pues OpenMP se encuentra dentro del modelo.



PIPELINE CI/CD CON JENKINS

09

Estado del servicio (Health Check)

Estado: **Healthy** ✓

Modelo cargado: Sí

Ver JSON: </health?raw=1>

[Volver al inicio](#)

- ✓ Checkout SCM 5.2s
- ✓ Checkout 3.2s
- ✓ Install Python Dependencies 15s
- ✓ Run Tests 17s
- ✓ Train Model 33s
- ✓ Build Docker Image 9.2s
- ✓ Test Container 0.72s
- ✓ Deploy 1.2s
- ✓ Post Actions 0.39s

PIPELINE CI/CD CON JENKINS

09

API de Predicción de Morosidad - Norte Andino SAC

Interfaz visual para pruebas rápidas



Predicción (POST)

Formulario para probar la predicción del modelo

[Ir al formulario](#)



Métricas del modelo (GET)

[Ver métricas](#)



Estado del servicio (GET)

[Ver estado](#)



Ejemplo en cURL

```
curl -X POST http://localhost:5000/predict \  
-H "Content-Type: application/json" \  
-d '{  
  "monto_original": 5000,  
  "monto_actual": 3000,  
  "ratio_deuda": 0.6,  
  "dias_desde_vencimiento": 45  
}'
```

CONCLUSIONES

El trabajo concluyó con éxito que la implementación del pipeline CI/CD con Docker y Jenkins, combinada con la paralelización por OpenMP, es clave para la optimización de procesos intensivos.

Los resultados demostraron la eficiencia de la solución al lograr un Speedup de 2.97 y una reducción del 66.36% en el tiempo de procesamiento, superando el objetivo del 60%.





**GRACIAS POR
SU ATENCIÓN**

BIBLIOGRAFÍA

- Aldinucci, M. et al. (2021). Practical Parallelization of Scientific Applications with OpenMP, OpenACC and MPI. Journal of Parallel and Distributed Computing. doi: 10.1016/j.jpdc.2021.05.017
- Monteiro, A. M. & Santos, A. A. F. (2023). Parallel computing in finance for estimating risk-neutral densities through option prices. Journal of Parallel and Distributed Computing. doi: 10.1016/j.jpdc.2022.11.010
- Oancea, B. (2014). Parallel computing in economics: An overview of the software frameworks. MPRA. <https://mpra.ub.uni-muenchen.de/72039/>
- Pawlak, W. M. (2021). Accelerated financial algorithms: Derivative pricing and risk management applications. Ph.D. Thesis, University of Copenhagen. <https://di.ku.dk/english/research/phd/phd-theses/2021/wojciechmikalpawlak-thesis.pdf>
- Susnjara, S. & Smalley, I. (2024). What is the CI/CD pipeline. IBM Think. <https://www.ibm.com/think/topics/ci-cd-pipeline>
- Tovar Galarreta, J. C. (2024). Material del curso Computación de Alto Desempeño y Cloud Computing. Universidad del Pacífico.
- Tripathi, R. & Abhishek (2024). Automated credit card default prediction using MLOps on AWS: From pipeline development to real-time deployment. Proc. Int. Conf. TIACOMP. doi: 10.1109/TIACOMP64125.2024.00050