(IJACSA) International Journal of Advanced Computer Science and Applications,
Vol. 16, No. 4, 2025

Flexible Software Architecture for Genetic Data Processing in Alpaca Breeding Programs

Alfredo Gama-Zapata, Fernando Barra-Quipse, Elizabeth Vidal
Escuela Profesional De Ingeniería De Sistemas, Universidad Nacional De San Agustín De Arequipa, Arequipa, Perú

Abstract—**Improving alpaca fiber quality is an important objective in the textile industry. There are different kinds of techniques aimed to enhance breeding outcomes. This study proposes and validates a flexible software architecture for managing genetic information in alpaca breeding, integrating genomic selection methods. The proposed architecture consists of three components: 1) Input—capturing data from individual records, pedigree, phenotypic traits, fiber characteristics, genomic, and non-genomic information; 2) Processing—implementing statistical methods such as BLUP, GBLUP, and SSGBLUP, alongside inbreeding coefficient calculation and machine learning techniques; and 3) Output—generating reports for mating list proposals, estimated breeding values, and genetic evaluations. Designing a software architecture for genetic improvement in alpaca breeding programs could help software developers with maintainability, extensibility, and adaptability, considering different kinds of data sources for future advancements in alpaca breeding. This work shows the implementation and validation of software for an alpaca breeding program based on the proposed architecture.**

Keywords—Architecture; genomic selection; adaptability

I. INTRODUCTION

The purpose of alpaca breeding is to improve the textile properties of the fiber [1]. The textile industry seeks quality fibers measured by the fineness and the low variability of its diameter [2]. Recently, new ways to enhance textile production have emerged through genomic selection.

Genetic improvement in alpaca breeding is of great interest, as it aims to optimize productive traits such as fiber quality. Genomic selection refers to using genome-wide and dense markers for predicting breeding values (BV) and the subsequent selection of individuals [3], [4]. To achieve this objective, genetic improvement programs have incorporated advanced data analysis and genetic modeling technologies, driving the development of specialized computational tools.

In this context, software architecture plays a crucial role in building efficient systems for collecting, processing, and analyzing information that enables the genetic improvement of alpacas. Genetic models used in genomic selection for animal breeding require the integration of multiple data sources (pedigree, genotypes, phenotypes, environmental data, etc.), intensive statistical computations, and the delivery of reliable and reproducible results.

In software architecture, different approaches are used to manage large volumes of genetic information efficiently [5]. However, the application of architecture models in alpaca genetic improvement still presents an opportunity for the development of more specialized solutions tailored to this species' specific characteristics. While there have been advancements in software development for managing genetic improvement information in animals [6], [7] there is a lack of software architectures designed explicitly for alpaca genetic improvement, considering the species' unique characteristics and the interoperability between different data sources (genotypic, phenotypic, and environmental).

A flexible architecture would allow a prediction module to be replaced with a more advanced one, without redesigning the entire system. The main objective of this study is to propose and validate a flexible software architecture for managing alpaca genetic information, integrating genomic selection methods and data processing for software developers in this field.

II. BACKGROUND

A. Software Architecture

Software architecture is the structure that comprises software components, their externally visible properties, and the relationships between them. According to Garlan [8], identifying and documenting a software architecture allows other developers to adopt previous architectural structures as a starting point. A well-designed architecture ensures that a system meets key requirements such as maintainability, extensibility, and adaptability [9], these are key attributes that significantly influence the long-term success and usability of software systems [10].

Maintainability refers to the ease with which a software system can be modified to correct faults, improve performance, or adapt to a changed environment. A key factor influencing maintainability is modularity. Modularity refers to a well-structured architecture that allows components to be updated or replaced independently without affecting the entire system.

Extensibility is the capability of a software system to accommodate future growth by adding new features or components without significant rework. Adaptability is the ability of a software system to evolve in response to changing requirements or environments. This characteristic is crucial in today’s fast-paced technological landscape.

B. Genomic Selection

Genomic selection increases the rate of genetic improvement and reduces the cost of progeny testing by allowing breeders to preselect animals that inherited chromosome segments of greater merit [11]. A Single

Nucleotide Polymorphism (SNP) is a slight difference in the DNA sequence that varies between individuals. These differences act like genetic "markers", helping researchers to track which genes an animal has inherited. Without SNP markers, researchers only rely on pedigree-based selection, which assumes all siblings inherit the same genetics [12]. Computer algorithms and programs are needed to incorporate genomic data into genetic evaluations and to process the rapidly expanding numbers of SNP genotypes.

There are statistical methods widely used in genomic selection to predict genetic merit in livestock breeding: Best Linear Unbiased Predictor (BLUP) [13], Genomic Best Linear Unbiased Predictor (GBLUP) [12] and Single-Step Genomic Best Linear Unbiased Predictor (SSGBLUP) [14].

III. METHODOLOGY

A. Software Architecture

This study presents software architecture for flexible software development aimed at the genetic improvement of alpacas, emphasizing textile quality. The proposed architecture consists of three components (Fig. 1). Component A -Input: This component captures the necessary data for processing: Individual, Pedigree, Phenotypic, Fiber, Genomic, and Non- Genomic Information. Component B - Processing: Comprises five modules, three of which are statistical methods used in genomic selection: BLUP, GBLUP, and SSGBLUP; the fourth module calculates the INBREEDING COEFFICIENT, and the fifth module allows to include machine learning techniques in order to improve statistical methods performance. Component C - Output: Presents three kinds of reports: Mating List Proposal, Estimated Breeding Values, and Genetic Report.

B. Component A – Input

Component A considers six different types of input data, described below:

Individual: It considers the animal's ear tag, date of birth, species, breed, color, and sex.

Pedigree: It refers to an alpaca's recorded ancestry including information about its parents, grandparents, and other ancestors.

Phenotypic: In alpaca breeding, key measurable traits include [15]:

- Density: The amount of fiber per unit area on the animal's body. A denser fleece is typically associated with higher fiber yield and superior quality.
- Leg Coverage: The alignment and proportion of the front and hind legs assess whether the animal has a proper and functional stance.
- Head: The overall shape and appearance of the head, including ear positioning, facial profile, and bone structure. It is important for functional and health-related assessments.
- Balance: The overall symmetry and proportion of the alpaca's body, including the relationship between the trunk, legs, and neck. Well-balanced animals are healthier, more productive, and have higher market value.
- Crimp: The shape and uniformity of the curls in the fiber. This parameter is related to the elasticity and softness of the produced textiles. Well-defined, uniform curls are indicators of high-quality fiber.

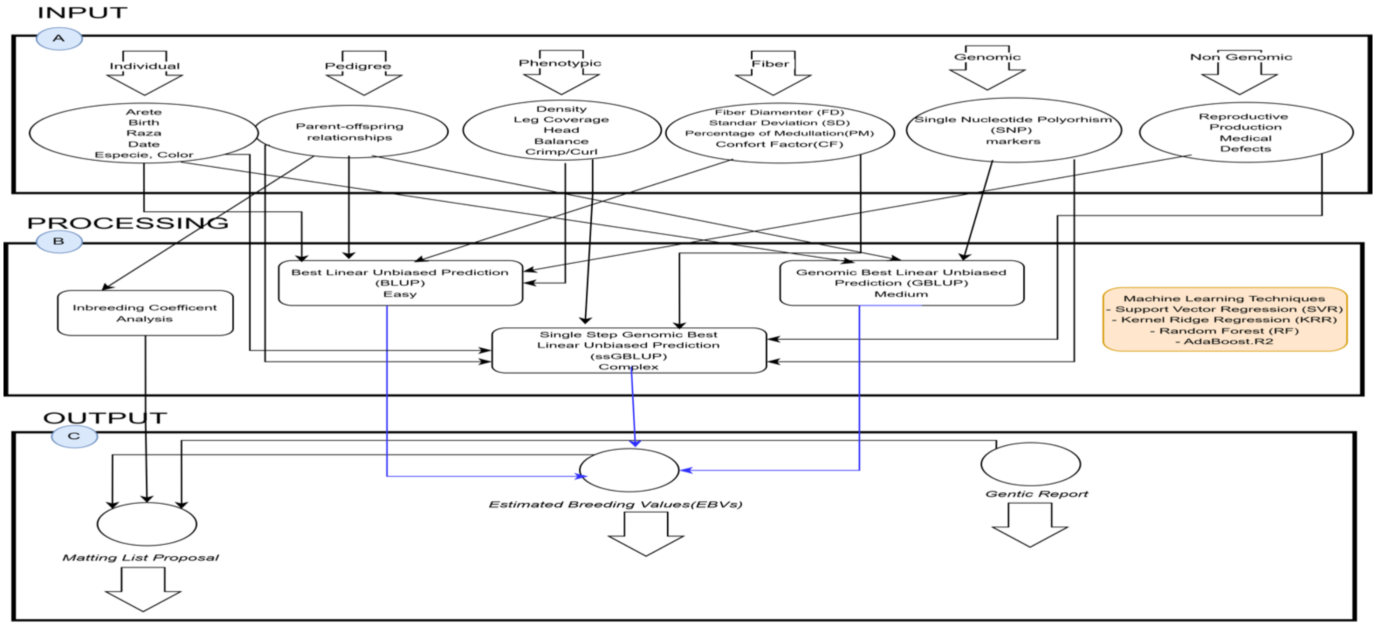


Fig. 1. Generic flexible software architecture for alpaca genetic data processing considering BLUP, GBLUP and SSGBLUP methods.

Fiber: These traits are recorded through direct measurements, such as fiber analysis after shearing. In alpaca breeding, key measurable traits include [16]:

- Fiber Diameter (FD): The thickness of the fiber in micrometers (µm).
- Standard Deviation of Fiber Diameter (SD): A measure of variability in fiber thickness. A lower SD reflects a more uniform fiber, essential for industrial processing.
- Percentage of Medullation (PM): This refers to the proportion of fibers with a hollow or partially hollow core, which affects fiber quality. Lower medullation percentages are preferred for fine textile applications.
- Micron (MIC): Represents fiber fineness, measured in microns (µm). A lower micron count indicates finer fiber, which is preferred in premium markets.
- Comfort Factor (CF): The percentage of fibers in the sample with a diameter of 30 µm or less, determining how "comfortable" the fiber feels against human skin. A higher CF indicates a lower likelihood of irritation.
- Coefficient of Variation (CV): The ratio of SD to the average fiber diameter, expressed as a percentage. It reflects the relative consistency of fiber diameter, with lower CV values being preferred.
- Average Medullated Fiber Diameter (MFD): Provides additional insights into fiber quality, as coarse and medullated fibers can reduce commercial value.

Genomic: This data refers to an organism's DNA information. In animal breeding, this data is used to find genes that influence important traits like fiber quality in alpacas. A Single Nucleotide Polymorphism (SNP) is a slight difference in the DNA sequence that varies between individuals. These differences act like genetic “markers,” helping us track which genes an animal has inherited.

Non-genomic: This data refers to a) reproductive: breeding, diagnostics, and births; b) production: type of fleece, weight, staple length; c) medical Information: treatments, diseases, and defects.

C. *Component B - Processing*

In genetic data processing for alpacas, it is important to have different methodological options such as BLUP, GBLUP, and SSGBLUP because each method offers distinct advantages depending on data availability and quality. BLUP relies on pedigree and phenotypic data, making it suitable when genomic information is limited or unavailable. GBLUP incorporates dense genomic markers to increase accuracy in the estimation of breeding values, which is particularly valuable when aiming for early selection and genetic gain. SSGBLUP integrates pedigree, phenotype, and genomic data into a single framework, allowing for a more comprehensive evaluation that maximizes the use of all available information. These methodological options allow breeders and researchers to tailor the approach based on their specific breeding goals, data structure, and computational resources, ensuring more precise and efficient genetic evaluations for the sustainable improvement of alpaca populations.

This component is the core of the proposed architecture, consisting of five modules that can be executed independently.

Three of them, BLUP (Best Linear Unbiased Prediction), GBLUP (Genomic Best Linear Unbiased Prediction), and SSGBLUP (Single-Step Genomic Best Linear Unbiased Prediction), are statistical methods used in animal breeding and genetic selection. These methods help estimate breeding values to improve desirable traits such as fiber quality in alpacas.

BLUP is a statistical method for estimating breeding values based on pedigree and phenotypic data. It assumes that genetic effects have a normal distribution and estimates genetic merit while adjusting for environmental effects. BLUP uses the relationship matrix called matrix A based on pedigree information. It assumes that the genetic values follow a linear mixed model. Also, it provides unbiased, minimum variance estimates of breeding values [13].

GBLUP is an extension of BLUP that incorporates genomic information using molecular markers (e.g., SNPs). Instead of the pedigree-based relationship matrix A, it uses a genomic relationship matrix G, built from SNP genotypes. GBLUP is more accurate than BLUP, as it captures actual genetic relationships rather than assuming them from pedigrees [12].

SSGBLUP is an improved version of GBLUP that combines pedigree, phenotype, and genomic data in a single step. It integrates both the traditional relationship matrix A and the genomic relationship matrix G into a combined relationship called matrix H. SSGBLUP allows for simultaneous evaluation of genotyped and non-genotyped animals [14].

Having this three statistical model options is important because developers and stakeholders could face different scenarios. The BLUP method is the best choice if only pedigree and phenotype data (fiber records) are available. If a genomic dataset is available, GBLUP is the method to use. Finally, if a mix of genotyped and non-genotyped alpacas exists, SSGBLUP is the option to select.

Using Machine Learning Techniques (MLT) in the genomic prediction of animal reproductive traits allows for improved prediction accuracy [17], [18], [19]. In genomic prediction, machine learning regression methods such as Support Vector Regression (SVR) [20], Kernel Ridge Regression (KRR) [21], Random Forest (RF) [22], and AdaBoost.R2 [23] are increasingly used to model the complex, nonlinear relationships between high-dimensional genetic marker data and quantitative phenotypic traits. These methods are particularly valuable in animal breeding, where accurate prediction of breeding values based on genomic information enables more efficient selection of superior individuals, thereby accelerating genetic gain.

SVR utilizes kernel functions to model complex, nonlinear associations between genetic markers and traits of interest. In the context of genomic selection, SVR handles high-dimensional SNP datasets by projecting them into a higher-dimensional feature space where linear relationships can be identified. SVR allows to capture subtle genetic effects that contribute to phenotypic variation.

KRR combines kernel methods' flexibility with ridge regression's regularization strength. In animal breeding, KRR helps model additive and non-additive genetic effects using non-linear kernels (e.g., Gaussian, polynomial) to map marker data into a higher-dimensional space. KRR enhances the analysis of traits influenced by many small-effect loci, allowing accurate estimation of genomic breeding values.

RF is an ensemble learning method based on decision trees for capturing interactions and nonlinear effects between genetic markers. RF's robustness to noise and ability to handle significant input variables without feature selection makes it a good option for animal breeding datasets, which often involve thousands of SNPs and limited sample sizes.

AdaBoost.R2 is particularly effective in emphasizing difficult-to-predict individuals, thereby refining predictions of phenotypic traits with heterogeneous genetic architectures. By adjusting sample weights based on previous errors, the algorithm focuses learning efforts on underrepresented or outlier phenotypes, improving the prediction of genomic breeding values and enhancing selection accuracy for traits with skewed or non-normal distributions.

Finally, the module Inbreeding Coefficient Analysis identifies common ancestors. This module analyzes the alpaca's pedigree chart to identify any ancestors on the sire's (father's) and dam's (mother's) sides. The most common method is Wright's Equation [24] shown in Eq. (1). The coefficient of consanguinity F, also known as the inbreeding coefficient, measures the probability that an individual has inherited two alleles at a given locus from a common ancestor. It is crucial in alpaca breeding to avoid excessive inbreeding, which can lead to genetic defects and reduced vigor.

$$F_\chi = \sum (\tfrac{1}{2})^{n_1+n_2+1}(1+F_A) \quad (1)$$

where,

F_χ = Inbreeding coefficient of the individual (X).

n_1 =Number of generations between the common ancestor and the sire.

n_2 = Number of generations between the common ancestor and the dam.

F_A = Inbreeding coefficient of the common ancestor (if unknown, assume 0).

$\sum$ = Summation over all common ancestors.

D. Component C – Output

This component presents tree kind of outputs.

The Estimated Breeding Value (EBV) is a numerical prediction of an alpaca’s genetic potential for a specific trait. EBVs help breeders select animals that pass desirable genetic traits to their offspring, improving overall herd quality.

The Genetic Report provides three items: a) pedigree analysis: ancestry verification, inbreeding coefficient; b) performance data: Measured traits such as fleece, growth, and reproduction; c) Genomic information: SNP markers, parentage confirmation.

The Mating List Proposal is a structured plan that suggests the optimal pairings of male and female alpacas to achieve specific breeding objectives. It maximizes genetic progress, improves desirable traits, and minimizes inbreeding while ensuring herd sustainability.

IV. RESULTS AND DISCUSSION

A. Implementation

The software developed is part of the Pacomarca Project [25] a research and genetic improvement program for alpacas in Peru. Its main objective is to manage and analyze genetic and phenotypic data to improve the quality of alpaca fiber, optimize selection programs, and maximize reproductive efficiency. The software focuses on pedigree and genealogical data registration, phenotypic data analysis, genotyping, and genetic evaluation, as well as simulation and prediction of genetic improvement.

Fig. 2 shows the architecture for the developed software. The Input component considers only five type of input data: the alpaca's individual, pedigree, phenotypic, fiber, and non- genomic information (Fig. 3).

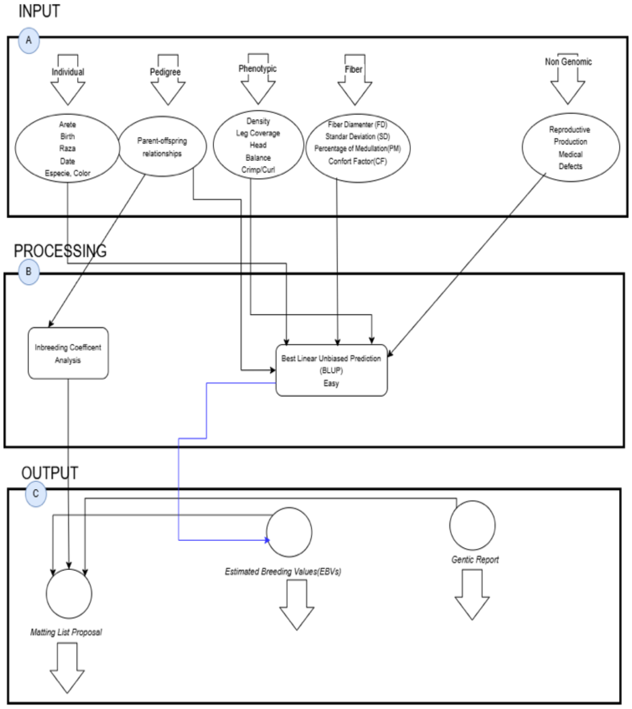
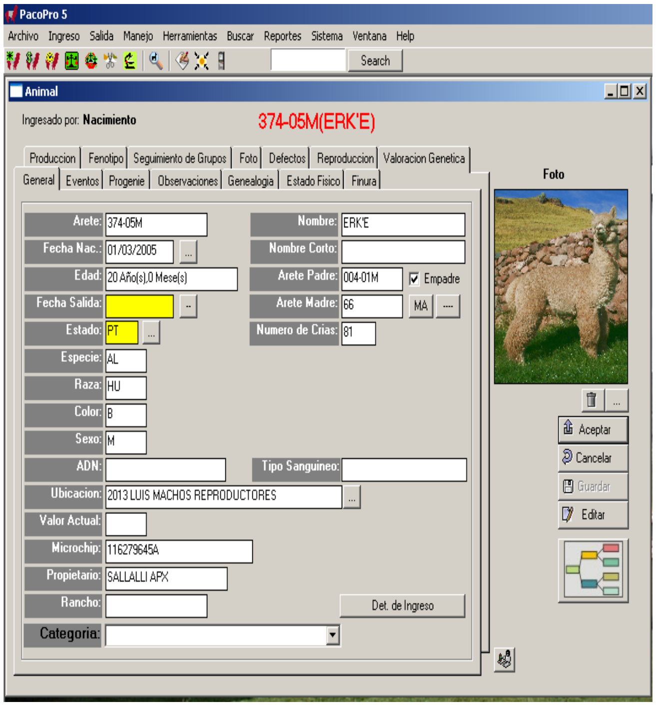


Fig. 2. Software architecture for alpaca breeding – Pacomarca project. It implements BLUP method and inbreeding coefficient analysis.

Processing component implements the BLUP module for genetic processing and the Inbreeding Analysis module. We show an implementation in Python. Fig. 4 shows the loading of pedigree, phenotype, and SNP data. Fig. 5 shows the processing of the matrices. Fig. 6 shows the processing of the BLUP method in line 19. Although the Pacomarca project has implemented the BLUP method only, the code for implementing GBLUP and SSGBLUP is also presented in lines 22 and 25, respectively. Fig. 7 shows the code for the Inbreeding Coefficient Analysis. Finally, Fig. 8 shows the result of the Estimated Breeding Value.

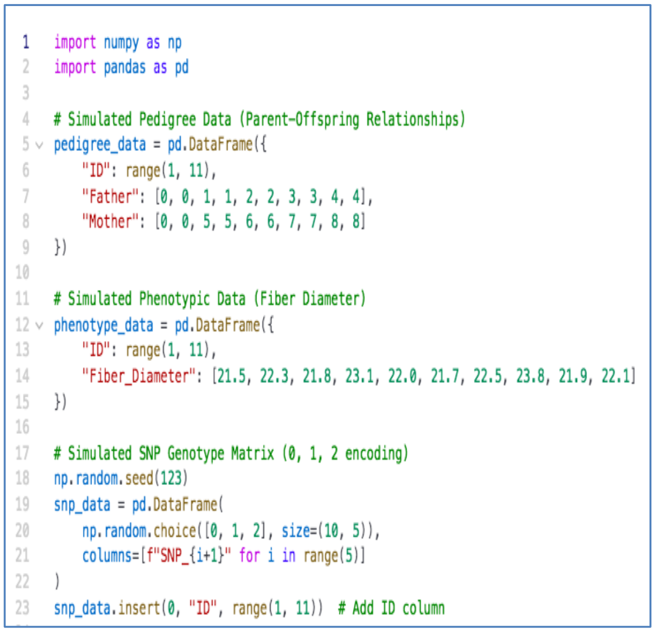


Ingresado por: **Nacimiento**

374-05M(ERK'E)

Arete: 374-05M
Nombre: ERK'E
Fecha Nac.: 01/03/2005
Nombre Corto:
Edad: 20 Año(s),0 Mese(s)
Arete Padre: 004-01M ☑ Empadre
Fecha Salida:
Arete Madre: 66
Estado: PT
Numero de Crias: 81
Especie: AL
Raza: HU
Color: B
Sexo: M
ADN:
Tipo Sanguineo:
Ubicacion: 2013 LUIS MACHOS REPRODUCTORES
Valor Actual:
Microchip: 116279645A
Propietario: SALLALLI APX
Rancho:
Categoria:

Fig. 3. Input component: software's main interface - individual, pedigree, phenotypic, fiber, and non-genomic information.



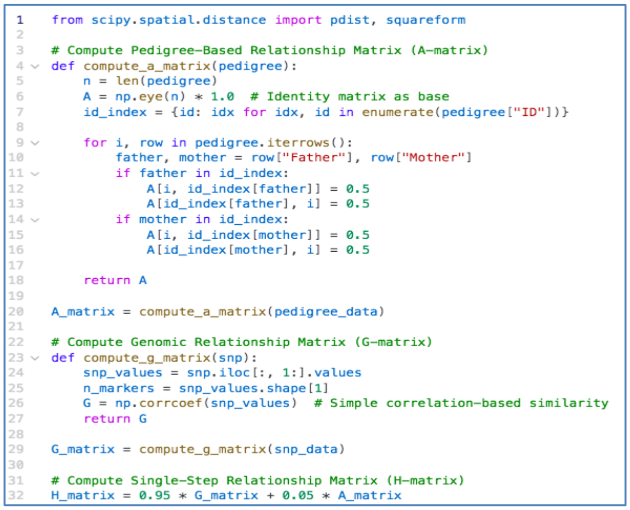
```python
import numpy as np
import pandas as pd

# Simulated Pedigree Data (Parent-Offspring Relationships)
pedigree_data = pd.DataFrame({
    "ID": range(1, 11),
    "Father": [0, 0, 1, 1, 2, 2, 3, 3, 4, 4],
    "Mother": [0, 0, 5, 5, 6, 6, 7, 7, 8, 8]
})

# Simulated Phenotypic Data (Fiber Diameter)
phenotype_data = pd.DataFrame({
    "ID": range(1, 11),
    "Fiber_Diameter": [21.5, 22.3, 21.8, 23.1, 22.0, 21.7, 22.5, 23.8, 21.9, 22.1]
})

# Simulated SNP Genotype Matrix (0, 1, 2 encoding)
np.random.seed(123)
snp_data = pd.DataFrame(
    np.random.choice([0, 1, 2], size=(10, 5)),
    columns=[f"SNP_{i+1}" for i in range(5)]
)
snp_data.insert(0, "ID", range(1, 11))  # Add ID column
```

Fig. 4. Processing component: Loading data.

B. Validation

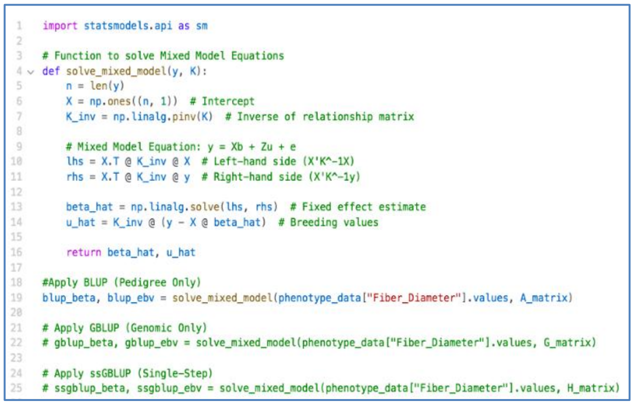
The software was validated with four experts from the alpaca breeding community in Peru and Spain. The instrument used was the User Experience Questionnaire (UEQ), which measures six dimensions: a) Attractiveness refers to the overall impression of the software. b) Perspicuity refers to whether the system is easy to get familiar with and use. c) Efficiency refers to whether users can complete tasks without effort. d) Dependability: assesses whether the user controls the interaction. e) Stimulation: evaluates whether the software is exciting and motivating. f) Novelty: considers whether the software design is creative and captures users' interest [26].

The scale ranges from -3 (terribly bad) to +3 (extremely good), with values between -0.8 and 0.8 representing a more or less neutral evaluation. Values above 0.8 indicate a positive assessment, while values below -0.8 indicate a negative evaluation. Table I presents the values of the six dimensions.



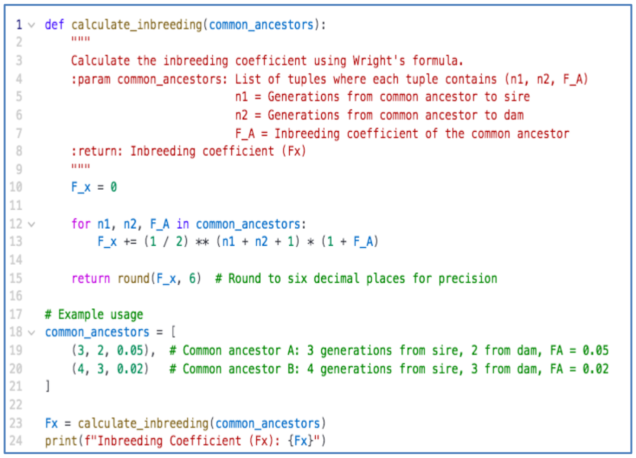
```python
from scipy.spatial.distance import pdist, squareform

# Compute Pedigree-Based Relationship Matrix (A-matrix)
def compute_a_matrix(pedigree):
    n = len(pedigree)
    A = np.eye(n) * 1.0  # Identity matrix as base
    id_index = {id: idx for idx, id in enumerate(pedigree["ID"])}

    for i, row in pedigree.iterrows():
        father, mother = row["Father"], row["Mother"]
        if father in id_index:
            A[i, id_index[father]] = 0.5
            A[id_index[father], i] = 0.5
        if mother in id_index:
            A[i, id_index[mother]] = 0.5
            A[id_index[mother], i] = 0.5

    return A

A_matrix = compute_a_matrix(pedigree_data)

# Compute Genomic Relationship Matrix (G-matrix)
def compute_g_matrix(snp):
    snp_values = snp.iloc[:, 1:].values
    n_markers = snp_values.shape[1]
    G = np.corrcoef(snp_values)  # Simple correlation-based similarity
    return G

G_matrix = compute_g_matrix(snp_data)

# Compute Single-Step Relationship Matrix (H-matrix)
H_matrix = 0.95 * G_matrix + 0.05 * A_matrix
```

Fig. 5. Processing component: compute relationships matrix.



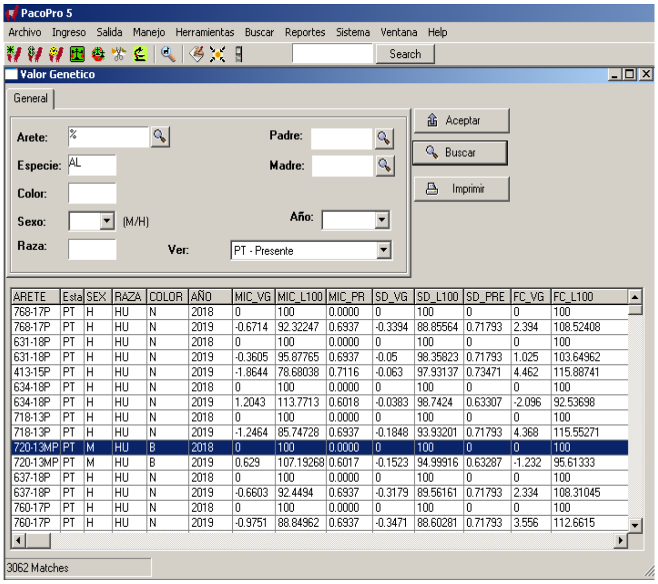
```python
import statsmodels.api as sm

# Function to solve Mixed Model Equations
def solve_mixed_model(y, K):
    n = len(y)
    X = np.ones((n, 1))  # Intercept
    K_inv = np.linalg.pinv(K)  # Inverse of relationship matrix

    # Mixed Model Equation: y = Xb + Zu + e
    lhs = X.T @ K_inv @ X  # Left-hand side (X'K^-1X)
    rhs = X.T @ K_inv @ y  # Right-hand side (X'K^-1y)

    beta_hat = np.linalg.solve(lhs, rhs)  # Fixed effect estimate
    u_hat = K_inv @ (y - X @ beta_hat)  # Breeding values

    return beta_hat, u_hat

#Apply BLUP (Pedigree Only)
blup_beta, blup_ebv = solve_mixed_model(phenotype_data["Fiber_Diameter"].values, A_matrix)

# Apply GBLUP (Genomic Only)
# gblup_beta, gblup_ebv = solve_mixed_model(phenotype_data["Fiber_Diameter"].values, G_matrix)

# Apply ssGBLUP (Single-Step)
# ssgblup_beta, ssgblup_ebv = solve_mixed_model(phenotype_data["Fiber_Diameter"].values, H_matrix)
```

Fig. 6. Processing component: Solving BLUP/ GBLUP/ SSGBLUP.



```python
def calculate_inbreeding(common_ancestors):
    """
    Calculate the inbreeding coefficient using Wright's formula.
    :param common_ancestors: List of tuples where each tuple contains (n1, n2, F_A)
                             n1 = Generations from common ancestor to sire
                             n2 = Generations from common ancestor to dam
                             F_A = Inbreeding coefficient of the common ancestor
    :return: Inbreeding coefficient (Fx)
    """
    F_x = 0

    for n1, n2, F_A in common_ancestors:
        F_x += (1 / 2) ** (n1 + n2 + 1) * (1 + F_A)

    return round(F_x, 6)  # Round to six decimal places for precision

# Example usage
common_ancestors = [
    (3, 2, 0.05),  # Common ancestor A: 3 generations from sire, 2 from dam, FA = 0.05
    (4, 3, 0.02)   # Common ancestor B: 4 generations from sire, 3 from dam, FA = 0.02
]

Fx = calculate_inbreeding(common_ancestors)
print(f"Inbreeding Coefficient (Fx): {Fx}")
```

Fig. 7. Processing component: Inbreeding coefficient analysis.



Arete: %
Especie: AL
Color:
Sexo: (M/H)
Raza:
Padre:
Madre:
Año:
Ver: PT - Presente

ARETE	Esta	SEX	RAZA	COLOR	AÑO	MIC_VG	MIC_L100	MIC_PR	SD_VG	SD_L100	SD_PRE	FC_VG	FC_L100
768-17P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
768-17P	PT	H	HU	N	2019	-0.6714	92.32247	0.6937	-0.3394	88.85564	0.71793	2.394	108.52408
631-18P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
631-18P	PT	H	HU	N	2019	-0.3605	95.87765	0.6937	-0.05	98.35823	0.71793	1.025	103.64962
413-15P	PT	H	HU	N	2019	-1.8644	78.68038	0.7116	-0.063	97.93137	0.73471	4.462	115.88741
634-18P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
634-18P	PT	H	HU	N	2019	1.2043	113.7713	0.6018	-0.0383	98.7424	0.63307	-2.096	92.53698
718-13P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
718-13P	PT	H	HU	N	2019	-1.2464	85.74728	0.6937	-0.1848	93.93201	0.71793	4.368	115.55271
720-13MP	PT	M	HU	B	2018	0	100	0.0000	0	100	0	0	100
720-13MP	PT	M	HU	B	2019	0.629	107.19268	0.6017	-0.1523	94.99916	0.63287	-1.232	95.61333
637-18P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
637-18P	PT	H	HU	N	2019	-0.6603	92.4494	0.6937	-0.3179	89.56161	0.71793	2.334	108.31045
760-17P	PT	H	HU	N	2018	0	100	0.0000	0	100	0	0	100
760-17P	PT	H	HU	N	2019	-0.9751	88.84962	0.6937	-0.3471	88.60281	0.71793	3.556	112.6615

3062 Matches

Fig. 8. Result component: Estimated breeding value.

TABLE I. USER EXPERIENCE RESULTS

Dimension	Mean	SD
Attractiveness	1.750	1.27
Perspicuity	1.125	2.06
Efficiency	2.063	0.89
Dependability	1.125	1.23
Stimulation	1.750	1.42
Novelty	1.375	0.60

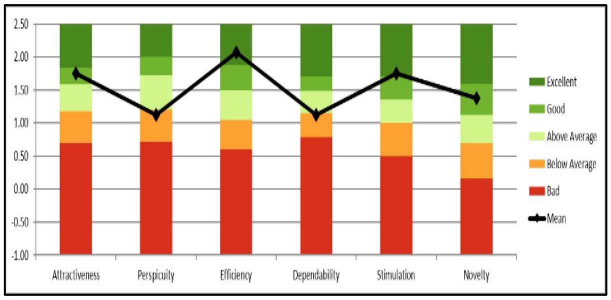


Fig. 9. Result of user experience questionnaire qualitative scale.

Fig. 9 shows a qualitative scale. The highest value is Efficiency, with an average score of 2.063, considered excellent. The second highest value was Stimulation, with an average score of 1.750, also considered excellent. Attractiveness obtained an average score of 1.750, rated as "good." The result reflects that users found the system visually appealing and pleasant to use. The score suggests that the system's design and aesthetics were positively received. Novelty obtained an average score of 1.375, rated as "good."

This result indicates that users perceived the system as innovative and featuring characteristics that captured their interest. Dependability obtained an average score of 1.125, below the average. This result may indicate that users did not feel complete control over the system, possibly due to a lack of clarity in the available options or intuitive handling. Finally, transparency obtained an average score of 1.125, which was also below the average. This result may reflect that users did not clearly understand how the system works or processes information.

C. Discussion

The main objective of this study was to propose and validate a flexible software architecture for managing alpaca genetic information, integrating genomic selection methods and data processing. A three-layer architecture has been created, where the Processing component consists of three genetic analysis modules and one module for analyzing the inbreeding coefficient. The architecture has been validated by implementing software that employs the BLUP method for genomic selection.

A specialized architecture for this type of system allows for flexibility when implementing solutions based on the available data, regarding using BLUP, GBLUP, or SSGBLUP methods. This type of architecture would allow for improved performance and adaptation to changing environments. Modularity would enable components to be updated or replaced independently without affecting the entire system. For example, while the processing methods were implemented in Python, depending on the developers' expertise, they could be implemented in ASReml-R [27]. ASReml-R is widely used in animal breeding and quantitative genetics because it efficiently handles large datasets.

Regarding extensibility, the proposed architecture allows the software system to accommodate future growth by adding new features or components without significant rework. For instance, this architecture could incorporate new genomic selection processing modules, such as BayesA or BayesB [28].

Additionally, this architecture could evolve in response to changing requirements or environments, a crucial characteristic in today's fast-paced technological landscape. For example, within the Processing component, it would be possible to introduce a new module integrating Machine Learning techniques to enhance the results obtained with BLUP, GBLUP, or SSGBLUP. Recent literature has already demonstrated progress in this area; Gianola et al. [29] presented enhancing genome-enabled prediction by bagging genomic BLUP. Wang et al. [18] implemented machine learning to improve the accuracy of genomic prediction of reproduction traits in pigs. Gianola et al. [30] presented Machine learning and genetic improvement of animals and plants: where are we? Santana et al. [31] presented a Genome-enabled cattle stability classification under a machine-learning framework.

The UEQ results showed that Efficiency had the highest score. Given the nature of the developed system, this result reinforces the idea that users obtained the expected outcomes.

This proposal highlights the importance of software architectures, especially in specific domains such as genetic improvement in alpaca breeding. This architectural model could serve as a foundation for use in other livestock species.

This work has some limitations; the main one is that only one implementation example was carried out, explicitly using the BLUP method. In future work, tests will be conducted using

GBLUP or SSGBLUP methods. A second limitation is the number of experts who tested the system. User access was relatively limited because this is a highly specialized field. Another limitation lies in the user experience evaluation, as the sample consisted of four experts in alpaca genetic improvement. This is due to the highly specialized nature of the area, which limits the ability to generalize the results obtained. In future work, it is intended to expand the usability evaluation of the proposed architecture with more experts in alpaca breeding from different regions.

V. CONCLUSION

This study addresses the design of flexible software architectures for genetic data processing in alpaca breeding programs. The proposal shows that it is possible to integrate specific input data, advanced algorithms, and predictive models based on BLUP, GBLUP, and SSGBLUP to optimize selection and breeding processes.

This work contributes to the field by providing a practical reference framework for designing and evaluating software architectures specialized in genomic analysis. It integrates computational methodologies that reduce processing time and improve the accuracy of predictive models. Additionally, it highlights the importance of using machine learning to efficiently manage large volumes of genomic information.

As an important conclusion, the study emphasizes that the evolution of software architectures for genomic selection should be guided by solid engineering principles aligned with the scientific community's and livestock producers' needs. For future work, it is suggested to explore the integration of artificial intelligence to enhance predictive model accuracy and develop automated systems that facilitate decision-making in alpaca breeding programs.

REFERENCES

[1] J. P. Gutiérrez, F. Goyache, A. Burgos, and I. Cervantes, “Genetic analysis of six production traits in Peruvian alpacas,” *Livest. Sci.*, vol. 123, no. 2–3, pp. 193–197, 2009.

[2] J. P. Gutiérrez, L. Varona, A. Pun, R. Morante, A. Burgos, I. Cervantes and M. A. Pérez-Cabal, “Genetic parameters for growth of fiber diameter in alpacas,” *J. Anim. Sci.*, vol. 89, no. 8, pp. 2310–2315, 2011.

[3] T. H. Meuwissen, B. J. Hayes, and M. Goddard, “Prediction of total genetic value using genome-wide dense marker maps,” *Genetics*, vol. 157, no. 4, pp. 1819–1829, 2001

[4] B. Mancisidor et al., “ssGBLUP method improves the accuracy of breeding value prediction in Huacaya Alpaca,” *Animals*, vol. 11, no. 11, p. 3052, 2021.

[5] V. Gancheva and I. Georgiev, “Software architecture for adaptive in silico knowledge discovery and decision making based on big genomic data analytics,” in *AIP Conference Proceedings*, vol. 2172, no. 1, AIP Publishing, Nov. 2019.

[6] S. Purcell et al., “PLINK: a tool set for whole-genome association and population-based linkage analyses,” *Am. J. Hum. Genet.*, vol. 81, pp. 559–575, 2007. [Online]. Available: https://doi.org/10.1086/519795.

[7] W. H. Lopez, “Reproductive biotechnologies in domestic South American camelids as alternatives for genetic improvement,” *Arch. Latinoam. Prod. Anim.*, vol. 23, no. 1–2, 2015.

[8] D. Garlan, “Software architecture: a travelogue,” in *Future Softw. Eng. Proc.*, 2014, pp. 29–39.

[9] D. Garlan, “Software architecture: a roadmap,” in *Proc. Conf. Future Softw. Eng.*, May 2000, pp. 91–101.

[10] M. Shaw and D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, 1996.

[11] T. Meuwissen, “Genomic selection: marker assisted selection on a genome wide scale,” *J. Anim. Breed. Genet.*, vol. 124, no. 6, pp. 321–322, 2007.

[12] P. M. VanRaden, “Efficient methods to compute genomic predictions,” *J. Dairy Sci.*, vol. 91, no. 11, pp. 4414–4423, 2008.

[13] C. R. Henderson, “Best linear unbiased estimation and prediction under a selection model,” *Biometrics*, vol. 31, no. 2, pp. 423–447, 1975.

[14] A. Legarra, O. F. Christensen, I. Aguilar, and I. Misztal, “Single Step, a general approach for genomic selection,” *Livest. Sci.*, vol. 166, pp. 54–65, 2014.

[15] R. Morante, A. Burgos, and J. P. Gutiérrez, “Producing alpaca fibre for the textile industry,” in *Fibre Production in South American Camelids and Other Fibre Animals*, Wageningen Academic, 2011, pp. 35–40.

[16] A. Cruz, R. Morante, J. P. Gutiérrez, R. Torres, A. Burgos, and I. Cervantes, “Genetic parameters for medullated fiber and its relationship with other productive traits in alpacas,” *Animal*, vol. 13, no. 7, pp. 1358–1364, 2019.

[17] J. Wang et al., “Enhancing genomic prediction accuracy of reproduction traits in Rongchang pigs through machine learning,” *Animals*, vol. 15, no. 4, p. 525, 2025.

[18] X. Wang et al., “Using machine learning to improve the accuracy of genomic prediction of reproduction traits in pigs,” *J. Anim. Sci. Biotechnol.*, vol. 13, no. 1, p. 60, 2022.

[19] R. Su et al., “Genomic selection in pig breeding: comparative analysis of machine learning algorithms,” *Genet. Sel. Evol.*, vol. 57, no. 1, p. 13, 2025.

[20] M. Awad and R. Khanna, “Support vector regression,” in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Eds. Cham: Springer, 2015, pp. 67–80.

[21] C. Diao, Y. Zhuo, R. Mao, W. Li, H. Du, L. Zhou, and J. Liu, “Weighted Kernel Ridge Regression to Improve Genomic Prediction,” *Agriculture*, vol. 15, no. 5, p. 445, 2025.

[22] R. K. Sarkar, A. R. Rao, P. K. Meher, T. Nepolean, and T. Mohapatra, “Evaluation of random forest regression for prediction of breeding value from genomewide SNPs,” *Journal of Genetics*, vol. 94, pp. 187–192, 2015.

[23] D. L. Shrestha and D. P. Solomatine, “Experiments with AdaBoost. RT, an improved boosting scheme for regression,” *Neural Computation*, vol. 18, no. 7, pp. 1678–1710, 2006.

[24] S. Wright, “Coefficients of inbreeding and relationship,” *Am. Nat.*, vol. 56, no. 645, pp. 330–338, 1922.

[25] *Pacomarca*, 2025. [Online]. Available: https://pacomarca.com/es/.

[26] M. Schrepp, *User Experience Questionnaire Handbook*, 2015.

[27] D. G. Butler, B. R. Cullis, A. R. Gilmour, B. J. Gogel, and R. Thompson, ASReml-R Reference Manual Version 4 ASReml estimates variance components under a general linear mixed model by residual maximum likelihood (REML), Hemel Hempstead: VSN International Ltd, 2018, p. 188.

[28] D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick, “Extension of the Bayesian alphabet for genomic selection,” *BMC Bioinformatics*, vol. 12, pp. 1–12, 2011.

[29] D. Gianola, K. A. Weigel, N. Krämer, A. Stella, and C. C. Schön, “Enhancing genome-enabled prediction by bagging genomic BLUP,” *PLoS One*, vol. 9, no. 4, p. e91693, 2014.

[30] D. Gianola, J. Crossa, O. Gonzalez-Recio, and G. J. M. Rosa, “Machine learning and genetic improvement of animals and plants: where are we?,” in *Proc. 12th World Congr. Genet. Appl. Livest. Prod. (WCGALP)*, Wageningen Academic Publishers, Dec. 2022, pp. 1676–1679.

[31] T. E. Z. Santana et al., “Genome-enabled classification of stayability in Nellore cattle under a machine learning framework,” *Livest. Sci.*, vol. 260, p. 104935, 2022.