



CÉDULA:
1751292419

APELLIDO - NOMBRE:
Robinson Alessandro Tandazo Reyes

FECHA
14/01/2026

CASO DE ESTUDIO:

KGD-Rusia; en el proyecto ANTDRON2K25 los entomólogos rusos requieren que el sistema permita experimentar con hormigas y otorgarles super habilidades al comer Genomas (X = Asexual, XX = macho, XY = hembra). Marque su caso de estudio.

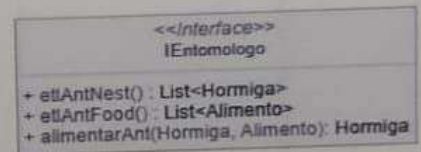
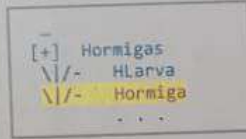
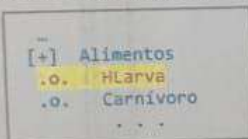
TipoHormiga	Alimento (vive)	Genomas →	SuperHabilidad	CASO (último Dígito Cédula)
- HLarva	Nectarívoros	-	-	todos
- HSoldado	Carnívoro	XX	superSaltar(...)	A 1,0
- HZángano	Omnívoro	XX	superVolar(...)	B 2,6
- HRastreadora	Insectívoro	XY	superRastreo(...)	C 3,7
- HReina	Nectarívoros	X	superReproductora(...)	D 4,8
- HObrera	Herbívoro	XY	superCortadora(...)	E 5,9

Storage (data lake):

- Agregar en storage/DataFiles/AntNest.txt → Hormiguero
- Agregar en storage/DataFiles/AntFood.txt → comida para hormigas

REQUERIMIENTOS:

- Al iniciar el sistema debe solicitar autenticación (usuario y contraseña; agregar "patmic", "123") con máximo 3 intentos. Si la autenticación es válida se debe presentar cédula y nombre completo.
- El entomólogo cosecha (según su caso de estudio) desde el storage hormigas y comida mediante un proceso ETL (Extract, Transform, Load-Database). Durante el proceso se debe mostrar Loading/Waiting por cada dato; Los datos válidos en muestran en color azul y se almacenan solo los datos de su caso de estudio; los datos con error se muestran en rojo.



- Implementar IEntomólogo y alimentar sus hormigas almacenadas en la base de datos, considerando:
 - preparar (Alimento)
 - La hormiga VIVE si come su alimento con o sin genoma caso contrario MUERE; el alimento comido se debe ser eliminado.
 - HLarva al comer su tipo de alimento con o sin genoma se transforma en su tipo de Hormiga y cambia de sexo acorde al genoma. Si el genoma corresponde a su caso de estudio se habilita la superHabilidad.
- Diseñar una propiedad a su Hormiga para la superHabilidad.
- Se debe garantizar que la acción de preparar (Alimento): Alimento sea realizado solo por el entomólogo o el entomólogo genetista. El entomólogo genetista prepara el alimento inyectando un genoma aleatoriamente (X, XX, XY) e imprimir: "[Preparado]-[tipoAlimento + Genoma]-"; mientras, el entomólogo prepara el alimento sin genomas e imprimir: "[Preparado]-[tipoAlimento]-"

OBLIGATORIO:

- Crear un proyecto java ApellidoNombre, incluir foto de la presente rúbrica y al finalizar el examen subirlo al TEAMS (Bimestre II / Prueba)
- El estándar para codificación será camelCase. Usar el prefijo con la primera letra/vocal de su apellido y nombre para atributos, variables, paquetes, clases, interfaces, atributos y métodos.

Ejemplo, si el alumno se llama: Pepe Lucho Pérez Suarez

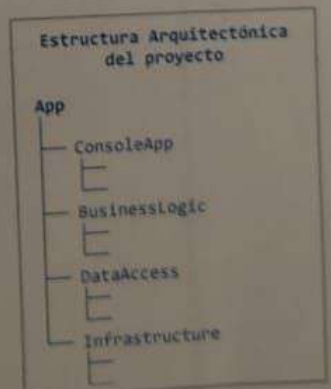
paquete: ppFanerozoico
métodos: void ppSetColor() {...}
int ppGetNumber() {...}
variables: int ppSuma;
atributos: public int ppNombre;

- El examen se anula si en la parte escrita (papel & lápiz) se detectar un intento de copia, entrega fuera de tiempo, uso de celular o tener abierto aplicaciones de chat o IA.
- Si la aplicación/diseño existen error / crash / no cumple con las instrucciones del presente: -0.5 puntos

CUESTIONARIO:

1. Realizar a mano el diagrama de caso de uso y MER (1 punto), diagrama de clases (2 puntos) y Estructura Arquitectónica del proyecto incluir capas, clases, interfaces, ... (1 punto)
2. Tomar foto a sus diagramas y realizar el código (2 Punto). Marcar con un comentario "Refactorización" los puntos que no coincidan con sus diseños.

Firma del alumno



Nombre: Robinson Alessandro Tandazo Reyes

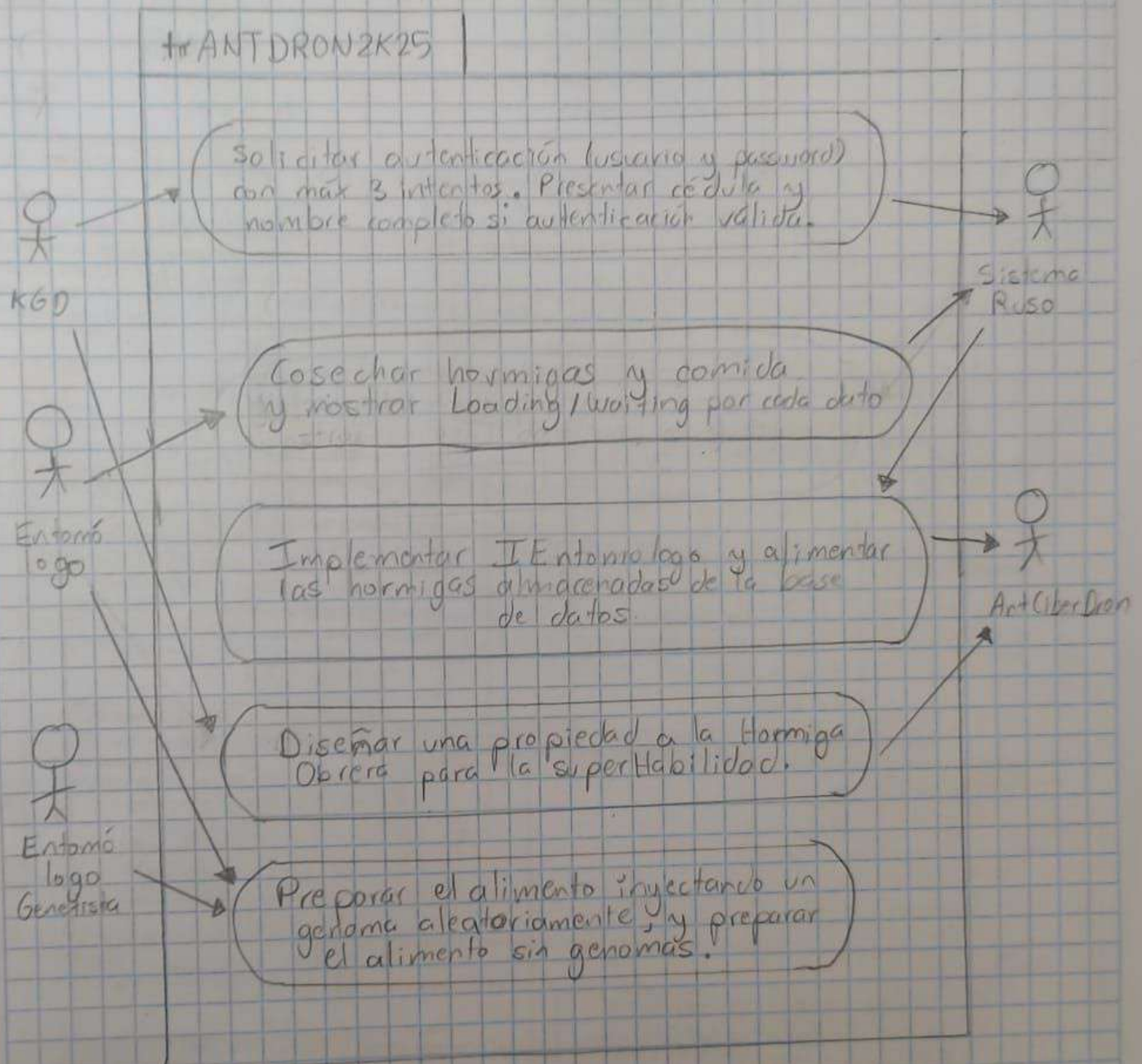
Fecha: 2026/04/14

Curso: GR2SW

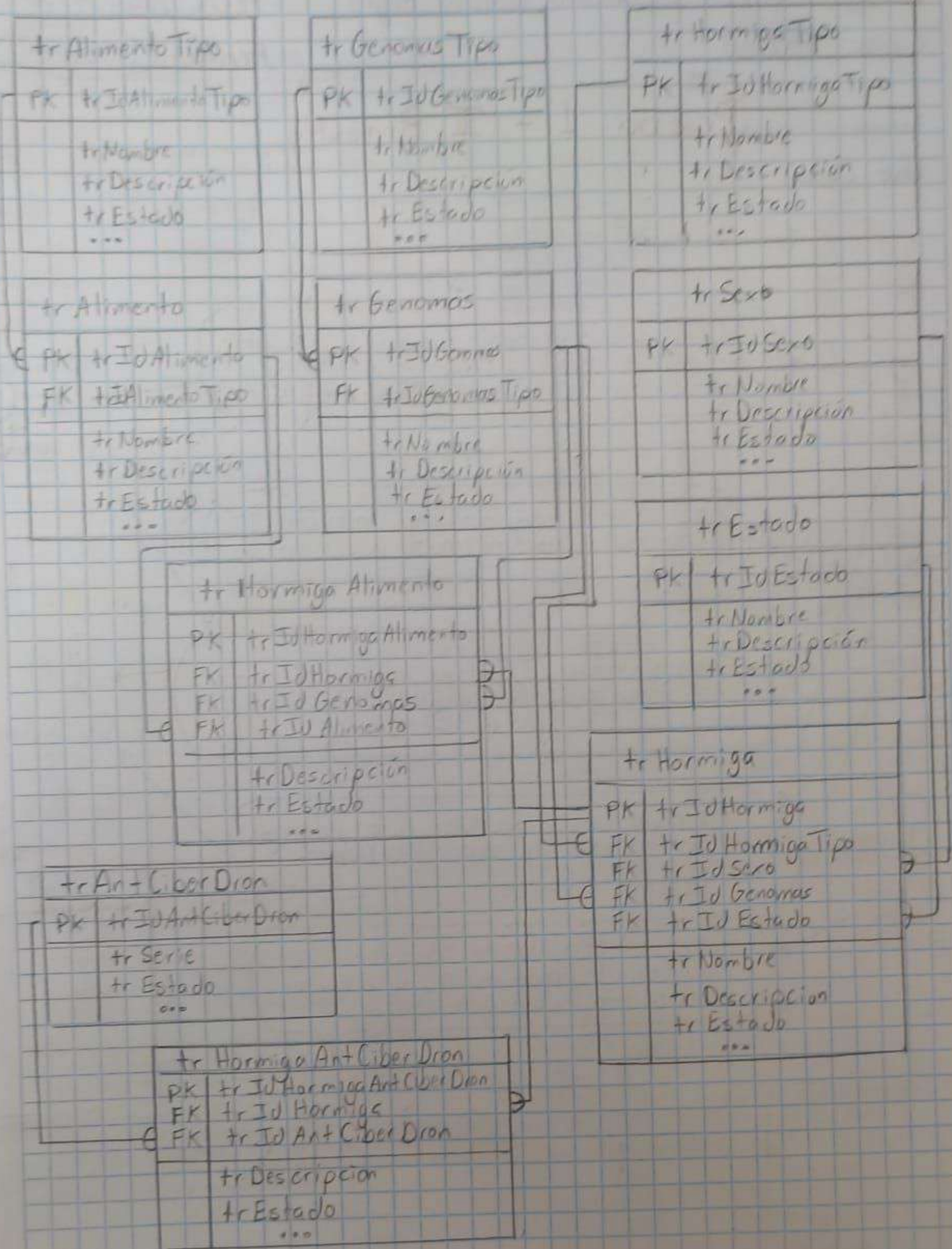
Firma: Robinson

Prueba 1 - II Bimestre

Use Case:



MER:



UML:

```

+tr KGD
- tr Cedula: String
- tr Nombre: String
- tr Usuario: String
- tr Clave: String

+ tr KGD (...)
+ show(): void
+ toString(): String
...
    
```

```

+tr Sistema Ruso

+ showLoading(): void
+ toString(): String
    
```

```

+tr AlimentoTipo
- tr Id AlimentoTipo: Int
- tr Nombre: String
- tr Descripción: String
...

+ tr AlimentoTipo()
+ tr AlimentoTipo(...)
+ toString(): String
...
    
```

```

+tr HormigaTipo
- tr Id HormigaTipo: Int
- tr Nombre: String
- tr Descripción: String
...

+ tr HormigaTipo()
+ tr HormigaTipo(...)
+ toString(): String
...
    
```

```

+tr Sexo
- tr Id Sexo: Int
- tr Nombre: String
- tr Descripción: String
...

+ tr Sexo()
+ tr Sexo(...)
+ toString(): String
...
    
```

```

+tr Estado
- tr Id Estado: Int
- tr Nombre: String
- tr Descripción: String
...

+ tr Estado()
+ tr Estado(...)
+ toString(): String
...
    
```

```

+tr GenomasTipo
- tr Id GenomasTipo: Int
- tr Nombre: String
- tr Descripción: String
...

+ tr GenomasTipo()
+ tr GenomasTipo(...)
+ toString(): String
...
    
```

+tr Interfaces

```

<< Interface >>
+tr IHormiga

+ tr Comer (Alimento, Genomas): Bool
+ tr Comer (Ingesta Nativa): Hormiga
    
```

```

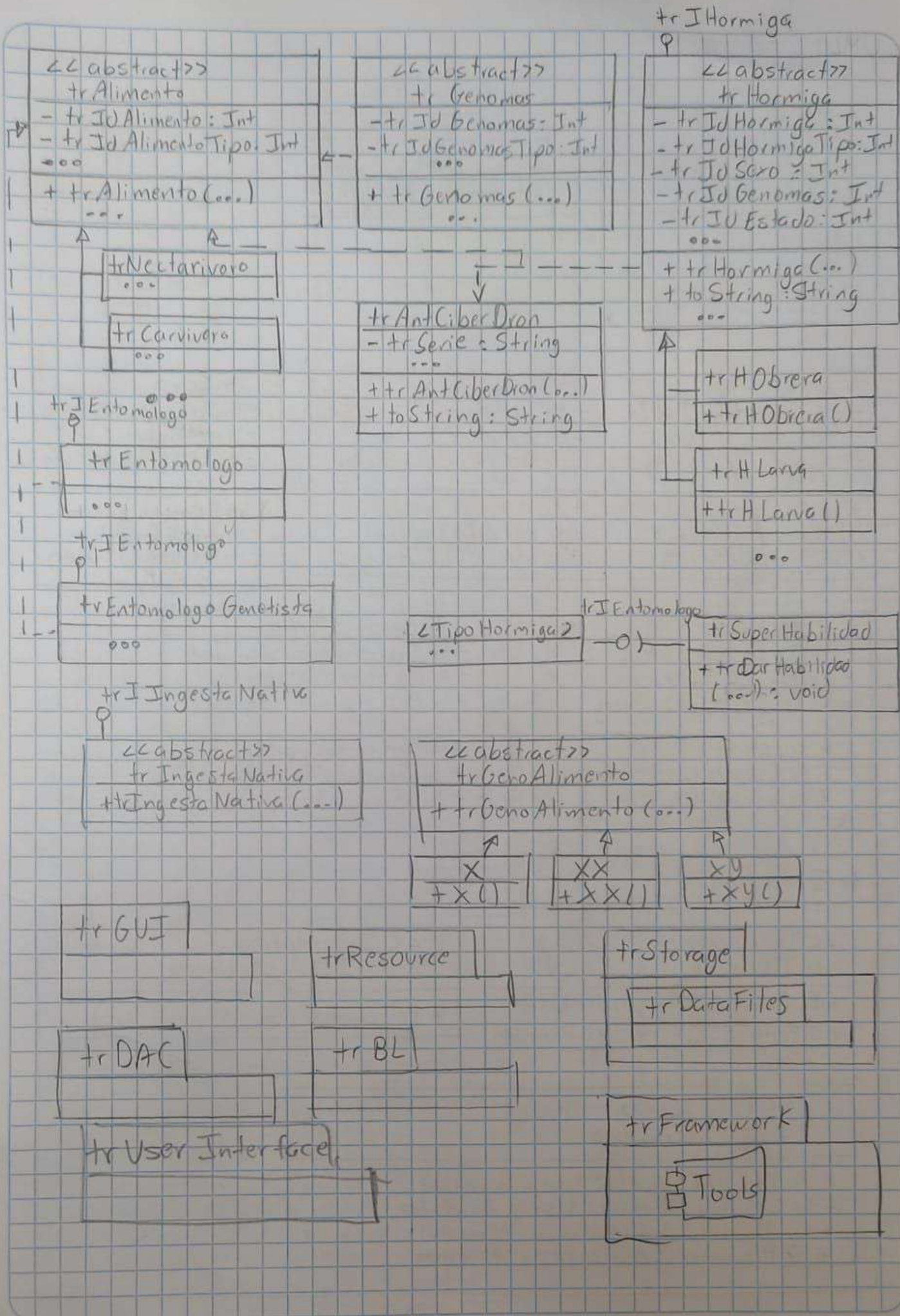
<< Interface >>
+tr IIngesta Nativa

+ tr Inyectar (Geno Alimento): bool
    
```

```

<< Interface >>
+tr IEntomólogo

+ tr Etl Ant Nest(): List < Hormiga >
+ tr Etl Ant Food(): List < Alimento >
+ tr Alimentar Ant (Hormiga, Alimento): Hormiga
+ tr Preparar (Alimento): Alimento
    
```

Estructura Arquitectónica del Proyecto

tr ANTDRON2K25

