

Londrina, PR / Julho, 2025

Automação na coleta e organização de dados experimentais: aplicação em estudos de biologia do percevejo *Euschistus heros* (Hemiptera: Pentatomidae)

Edson Hirose⁽¹⁾, Andressa Juvêncio da Silva Dias⁽²⁾, Jhon Henrique Wesnay Matheus Moreira Santos⁽³⁾, Leonardo José Motta Campos⁽⁴⁾

⁽¹⁾ Engenheiro-agrônomo, doutor em Entomologia, pesquisador da Embrapa Soja, Goiânia, GO.

⁽²⁾ Estudante de Agronomia, Universidade Federal de Goiás, Goiânia, GO. ⁽³⁾ Estudante de Agronomia, Centro Universitário de Goiás (UNIGOIÁS), Goiânia, GO. ⁽⁴⁾ Engenheiro-agrônomo, doutor em Biologia Vegetal, pesquisador da Embrapa Soja, Goiânia, GO. ⁽²⁾

Resumo - A confiabilidade na coleta e transcrição de dados é fundamental para garantir a validade dos resultados experimentais, especialmente em pesquisas que ainda utilizam formulários manuscritos, mais propensos a erros. O objetivo deste trabalho foi desenvolver um fluxo automatizado para coleta e organização de dados biológicos de percevejos, por meio da integração de tecnologias que assegurem rastreabilidade e confiabilidade ao processo. Foram estabelecidos protocolos padronizados de coleta de dados biológicos em laboratório e desenvolvidos três algoritmos interconectados em linguagem Python para reorganizar dos dados para análise estatística. A proposta foi validada em um bioensaio com *Euschistus heros* (Hemiptera: Pentatomidae), avaliando os efeitos de dietas à base de sementes de soja e amendoim sobre parâmetros biológicos na fase ninfal e adulta. Os resultados indicaram diferenças significativas entre as dietas, com maior longevidade e maior número de ovos por fêmea nas dietas contendo sementes de amendoim. O sistema automatizado proposto demonstrou ser uma alternativa eficiente para reduzir erros manuais e otimizar o processo de análise em laboratórios que conduzem bioensaios com insetos.

Termos para indexação: Cadernos de laboratório eletrônicos, algoritmos em Python, QR code, manejo integrado de pragas.

Automation in the collection and organization of experimental data: application in biological studies of the stink bug *Euschistus heros* (Hemiptera: Pentatomidae)

Abstract - Reliability in data collection and transcription is crucial to ensure the validity of experimental results, especially in studies that still use paper-based forms, which are more prone to errors. This study aimed to develop an automated workflow for collecting and organizing biological data on stink bugs by integrating technologies that ensure traceability and reliability. Standardized data collection protocols were established, and three Python algorithms were developed to systematize the data. The workflow was validated through in a bioassay using *Euschistus heros* (Hemiptera: Pentatomidae), evaluating the effects of soybean and peanut seed-based

Embrapa Soja

Rod. Carlos João Strass, s/n,
acesso Orlando Amaral
Caixa Postal 4006,
CEP 86085-981
Distrito de Warta
Londrina, PR
(43) 3371 6000
www.embrapa.br/soja
www.embrapa.br/fale-conosco/sac

Comitê Local de Publicações

Presidente

Roberta Aparecida Carnevalli

Secretária-executiva

Regina Maria Villas Bôas de Campos Leite

Membros

Clara Beatriz Hoffmann-Campo,
Claudine Dinali Santos Seixas,
Claudio Guilherme Portela de
Carvalho, Fernando Augusto
Henning, Leandro Eugênio
Cardamone Diniz, Liliane Márcia
Mertz-Henning, Maria Cristina
Neves de Oliveira e Norman
Neumaier

Edição executiva

Vanessa Fuzinatto Dall'Agnol

Revisão de texto

Regina Maria Villas Bôas de Campos Leite

Normalização

Valéria de Fátima Cardoso
(CRB- 9/1188)

Projeto gráfico

Leandro Sousa Fazio

Diagramação

Marisa Yuri Horikawa

Publicação digital: PDF

Todos os direitos reservados à Embrapa.

diets on biological parameters during the nymph and adult stages. The results showed significant differences between diets, with increased longevity and a higher number of eggs per female observed in individuals fed peanut seeds. The proposed system proved an efficient alternative for minimizing manual errors, and optimizing the data analysis process in laboratories conducting insect bioassays.

Index terms: electronic lab notebook, Python algorithms, QR code, integrated pest management.

Introdução

É consenso que humanos estão sujeitos a erros involuntários durante a coleta e digitação de dados. Infelizmente, falhas na entrada de dados podem ter efeitos devastadores sobre os resultados de uma pesquisa, podendo comprometer toda a análise estatística e, conseqüentemente, a validade da pesquisa (Barchard; Pace, 2011). As causas desses erros laboratoriais podem estar associadas a fatores psicológicos, como estresse ou distração, bem como a fatores ambientais, como má iluminação ou espaço físico inadequado de trabalho (Pereira et al., 2018). Embora existam técnicas para tais erros, compreender suas causas e corrigi-las é essencial para garantir a integridade dos resultados (Barchard; Pace, 2011).

Segundo Ley et al. (2019), os formulários de relato de caso em papel (CRFs – Case Report Forms) ainda são o método mais utilizado para coleta de dados em estudos de campo. Nessa abordagem, os dados são inicialmente registrados em papel e, posteriormente, digitalizados. Embora de fácil implementação, os sistemas baseados em registro em papel apresentam riscos de introdução de erros, tanto na coleta quanto na digitalização dos dados. No estudo conduzido por Ley et al. (2019), foram observadas discrepâncias de até 18% em dados contínuos coletados manualmente, em comparação com aqueles obtidos por meio da coleta eletrônica de dados (EDC – Electronic Data Collection). A EDC tornou-se uma alternativa amplamente aceita à coleta baseada em papel (PBDC – Paper-Based Data Collection), uma vez que permite verificações em tempo real da consistência lógica das informações registradas.

No âmbito da pesquisa científica, os Cadernos de Laboratório Eletrônicos (ELNs - Electronic Lab Notebook) já são amplamente conhecidos e oferecem benefícios como rastreabilidade parcial e preservação de dados experimentais. Eles proporcionam registros seguros e pesquisáveis de procedimentos e resultados, reduzindo o risco de

erros de transcrição e perda de informações (Kanza et al., 2017).

Entre os vários tipos de experimentos realizados em laboratórios de entomologia aplicada, destacam-se estudos da biologia dos insetos em diferentes condições bióticas e abióticas, que fundamentam o desenvolvimento do Manejo Integrado de Pragas (MIP). Esses estudos abrangem desde a construção de tabelas de expectativa de vida, utilizadas para analisar o desenvolvimento, sobrevivência e padrões de fecundidade, fundamentais para a compreensão da dinâmica populacional das espécies (Southwood, 1978), até a elucidação do papel nutricional de diferentes plantas hospedeiras no desenvolvimento dos percevejos (Panizzi; Lucini, 2017), além da formulação de novas técnicas para criação massal (Fortes et al., 2006; Mendoza et al., 2016).

Apesar das metodologias de bioensaio para avaliação alimentar de percevejos da família Pentatomidae já serem amplamente conhecidas e validadas (Tomacheski et al., 2019; Possebom et al., 2020), a condução de bioensaios laboratoriais extensos, voltados à avaliação simultânea de várias fontes nutricionais ou à análise de variações em parâmetros como temperatura e fotoperíodo, apresenta limitações operacionais. Estas incluem desde restrições de espaço físico, até a ocorrência de erros sistemáticos na coleta e transcrição manual dos dados.

Assim, à redução de erros humanos em laboratórios é essencial não apenas para a integridade dos resultados obtidos, mas também para a preservação da confiança nas conclusões científicas. Nesse contexto, o objetivo deste trabalho foi apresentar um fluxo automatizado para coleta e organização de dados de biologia de percevejos, por meio da integração de tecnologias que garantam confiabilidade e rastreabilidade, promovendo a transição para sistemas baseados em ELNs.

Material e Métodos

Procedência dos insetos

Posturas de *Euschistus heros* (Hemiptera: Pentatomidae) foram obtidas a partir da criação mantida na Embrapa Arroz e Feijão, localizada em Santo Antônio de Goiás, GO. As massas de ovos foram coletadas dessa criação e transferidas para placas de Petri (11 cm ϕ \times 2 cm de altura), com o fundo forrado com papel filtro e contendo uma vagem de feijão, para a manutenção das ninfas recém-emergidas. Ao atingirem o segundo instar, as ninfas foram individualizadas em gaiolas de bioensaio.

Gaiolas de bioensaio

Para avaliar os procedimentos de coleta de dados, foi realizado um bioensaio com três fontes alimentares: T1, sementes de soja - *Glycine max* (L.) Merr; T2, sementes de amendoim - *Arachis hypogaea* L.; e T3, sementes de soja e amendoim. Ninfas de segundo instar foram individualizadas

(n = 40 por tratamento) em gaiolas compostas por frascos plásticos de 80 mL (5 cm ϕ \times 6 cm de altura), adaptados para fornecimento de água *ad libitum*. Cada gaiola foi etiquetada com um QR Code e disposta em bandejas plásticas para facilitar o manuseio (Figura 1), sendo mantidas em sala climatizada a 26 ± 2 °C, com fotofase de 14 horas.

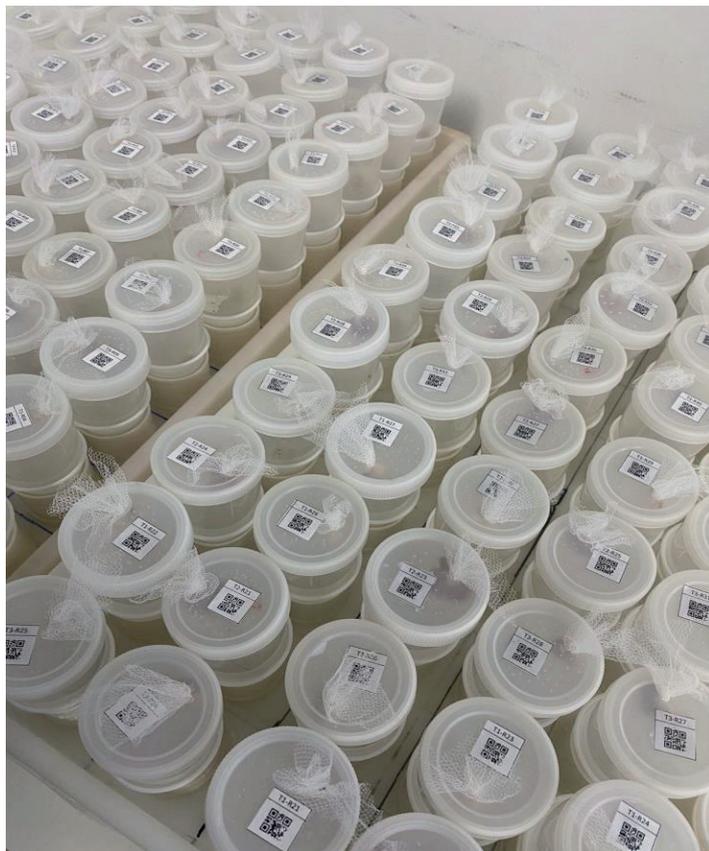


Figura 1. Bandejas com as gaiolas de bioensaio etiquetadas com QR codes. Embrapa Arroz e Feijão, abril de 2024.

Etiquetas com códigos de resposta rápida (QR Codes)

Foram utilizados três tipos de etiquetas para acompanhar as diferentes fases da biologia do percevejo (Figura 2). Para a fase jovem (do 2º instar até o primeiro dia da fase adulta), foi utilizada a etiqueta **modelo A**, codificada com seis dígitos no formato “T00R00”. O primeiro caractere, “T”, representa o tratamento, seguido por dois dígitos numéricos que permitem a codificação de até 100 tratamentos. O quarto caractere, “R”, representa a repetição, seguido por mais dois dígitos numéricos. Para a fase adulta (**modelo B**), a codificação utilizada foi composta por oito dígitos, no formato “CAT00R00”, sendo “CA” a identificação dos casais provenientes de cada tratamento, seguida pela codificação do tratamento e da repetição, conforme o mesmo padrão anterior. Para o rastreamento de cada postura e da emergência das ninfas, foi utilizada

uma terceira codificação (**modelo C**), no formato “ov000”, aplicada aos tubos de microcentrífuga de 1,5 mL.

Todos os QR Codes foram gerados utilizando uma Interface de Programação de Aplicações (API) do *QR Code Generator* (GOQR.ME, 2025), integrada a uma planilha eletrônica desenvolvida para a geração sequencial das etiquetas.

As etiquetas foram impressas em folhas de papel sulfite branco, tamanho A4, utilizando uma impressora a laser — o que elimina o risco de perda por molhamento — e, posteriormente, recortadas conforme as dimensões apropriadas. Em seguida, foram fixadas com cola branca atóxica nas gaiolas de bioensaio; no caso dos tubos de microcentrífuga, utilizou-se fita adesiva transparente para garantir melhor fixação.

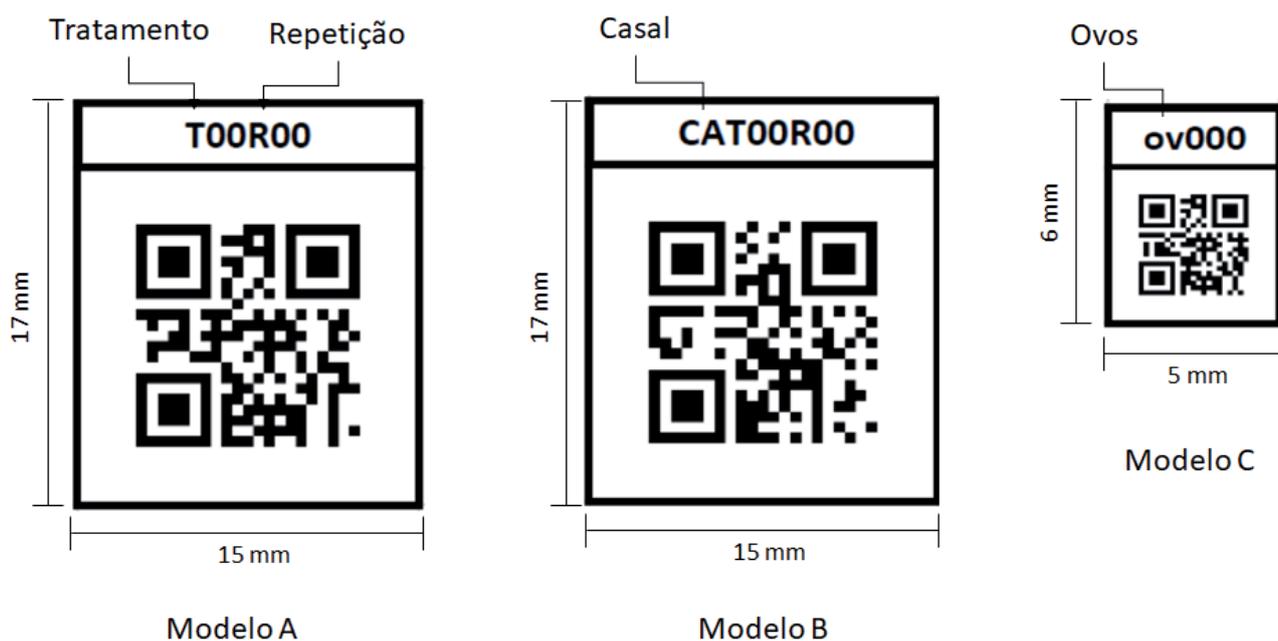


Figura 2. Etiquetas codificadas com QR codes para as gaiolas de biologia do percevejo *Euschistus heros* (Hemiptera: Pentatomidae) na fase ninfal (modelo A), fase adulta (modelo B) e para os tubos de microcentrifuga de 1,5 mL para as posturas (modelo C).

Sistemas para a leitura dos QR Codes

Planilhas eletrônicas padronizadas foram elaboradas utilizando o Google Docs, para armazenamento em nuvem, e o programa Excel Office 2019 – Microsoft, para armazenamento local. Em cada planilha, foram criadas cinco abas: “ninfa”, “adulto”, “casais”, “posturas” e “emergidos”, sendo cada uma preenchida em diferentes momentos ao longo do fluxo do experimento.

As planilhas foram manipuladas em um tablet com sistema Android 11 e em um notebook com sistema operacional Windows 10. Ambos os equipamentos foram utilizados em diferentes ocasiões e conectados a dois modelos de leitores 2D (QR Code): um com conexão sem fio via Bluetooth (modelo ND-59095 – marca Novavida com resolução de 640 x 480 pixels), utilizado com o tablet; e outro com conexão com fio via porta USB (modelo genérico, com resolução de 1280 x 800 pixels), utilizado com um notebook (Figura 3).



Figura 3. Leitores 2D utilizados para avaliar o sistema de QRcodes utilizadas nas biológicas de ninfas e adultos de *Euschistus heros* (Hemiptera: Pentatomidae) em dietas, à esquerda o modelo sem fio com conexão Bluetooth (sensor de 640 x 480 pixels) e a direita o leitor com conexão por porta USB (sensor de 1280 x 800 pixels).

Coleta dos dados biológicos na fase ninfal

As gaiolas foram avaliadas diariamente quanto à troca de instar, mortalidade das ninfas e emergência dos adultos. Nesta fase, os dados foram coletados utilizando a aba “ninfas” da planilha, conforme o seguinte procedimento: a cada avaliação, a primeira informação registrada era a data. Em cada avaliação das gaiolas, quatro situações eram possíveis: troca de instar, fuga ou perda do inseto, morte do inseto, emergência do adulto, sendo possível determinar o sexo. Para facilitar o registro dos eventos, foram impressas quatro imagens com QR Codes, cada uma

com a seguinte codificação: “MORTO”, “PERDIDO”, “MACHO” e “FEMEA”. Durante as avaliações, caso fossem observadas apenas exúvias (indicando troca de instar), apenas a etiqueta da gaiola era lida. Caso ocorresse uma das outras situações, a etiqueta da gaiola era lida primeiro, seguida da leitura do QR Code correspondente ao evento observado (Figura 4). Quando nenhuma das quatro situações era observada, não era necessária a leitura de nenhum código adicional. Todos os dados dessa fase foram registrados exclusivamente na aba “ninfas”.

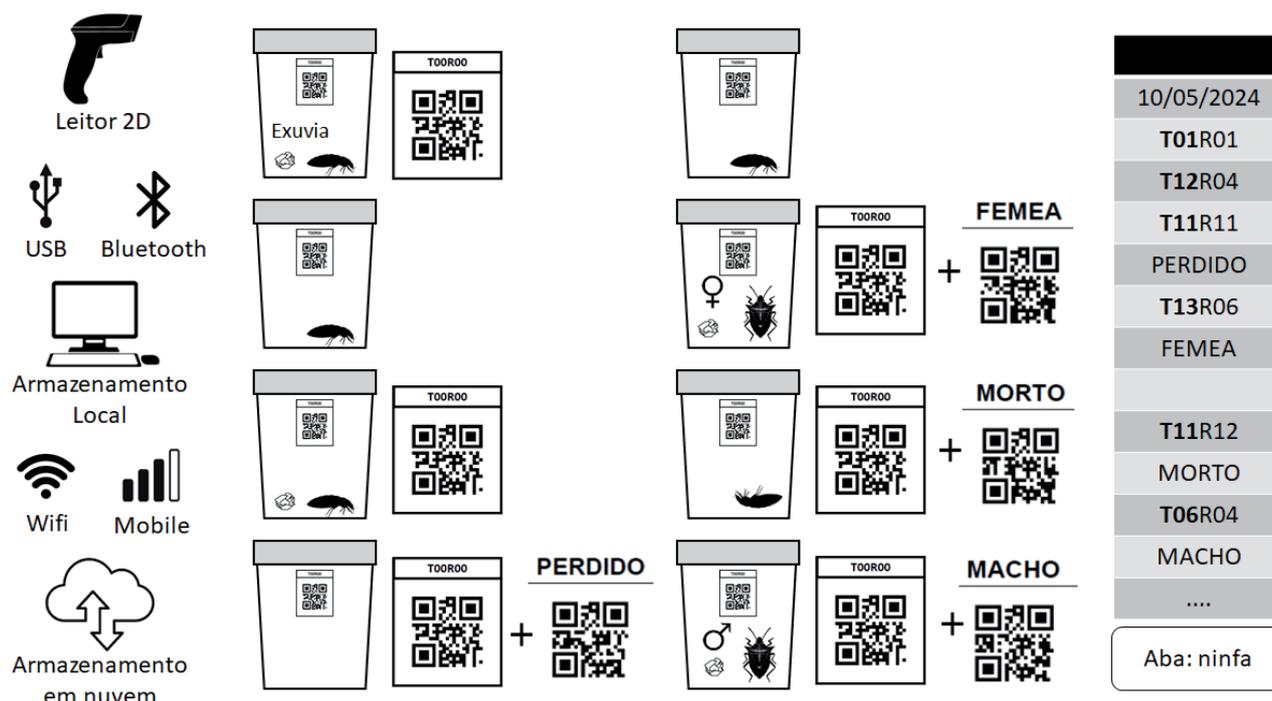


Figura 4. Esquema da coleta dos dados através de leitores 2D, e as possíveis sequências de entrada de dados na fase ninfal do percevejo *Euschistus heros* (Hemiptera: Pentatomidae).

Coleta da massa (mg) no primeiro dia da fase adulta

No primeiro dia da fase adulta, as gaiolas com os insetos foram levadas para pesagem, utilizando uma balança eletrônica analítica (modelo ATY 224R – marca Shimadzu). Esse modelo específico possui uma saída de dados USB tipo B, que permite a conexão com um computador por meio de entrada USB, transferindo automaticamente o peso registrado. O sistema utilizado para esse procedimento foi composto por: planilha eletrônica, computador, balança e leitor 2D (QR Code). O processo de pesagem seguiu os seguintes passos:

abrir a aba “adulto” na planilha eletrônica, inserir a data, ler o código da gaiola, verificar novamente o sexo do inseto, posicioná-lo na balança e, por fim, enviar o valor da massa diretamente para o computador. A sequência de entrada dos dados deve sempre começar com a leitura do código da gaiola. No entanto, a ordem de entrada dos demais dados (massa, sexo ou eventuais linhas vazias) não interfere na funcionalidade do sistema (Figura 5). Durante esse procedimento, são gerados dois dados redundantes — “data” e “sexo” — que podem ser utilizados para cruzamento e verificação da consistência dos registros.

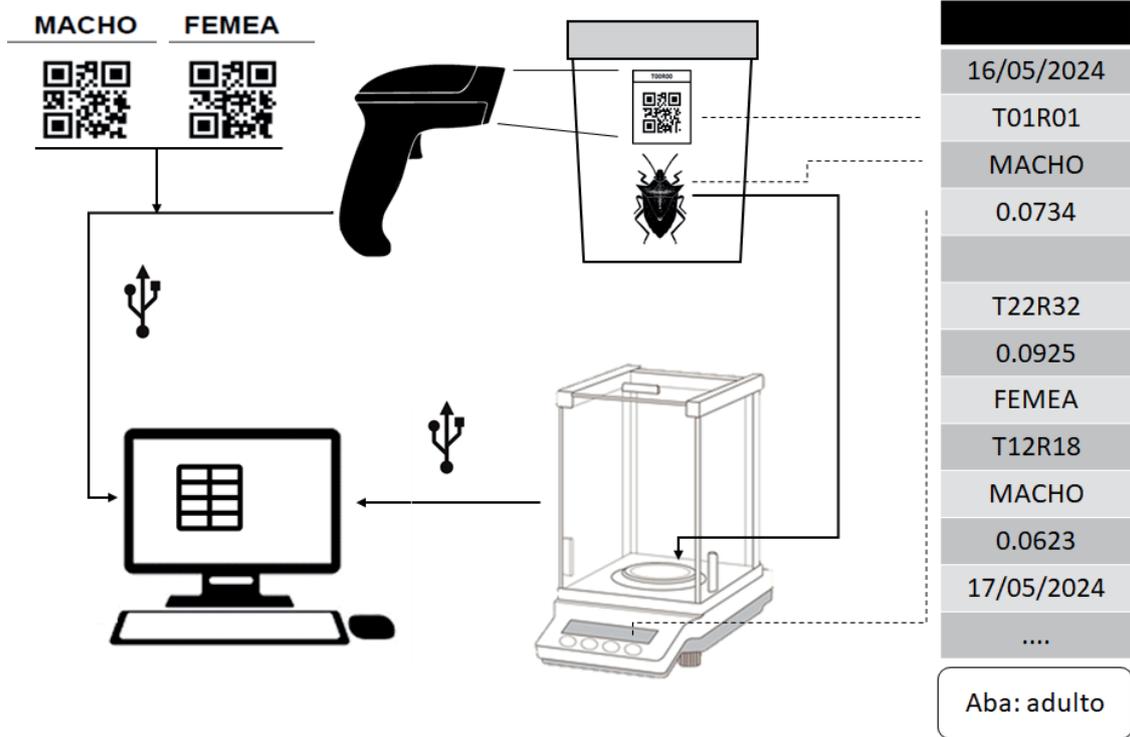


Figura 5. Esquema do sistema para coleta de dados da pesagem de perceijos.

Montagem das gaiolas de casais entre 24 e 48 horas após a emergência dos adultos

Adultos com até 48 horas de diferença na data de emergência, provenientes do mesmo tratamento alimentar, foram agrupados para formar casais. Uma nova gaiola de bioensaio, conforme descrito anteriormente, foi utilizada para acompanhar a fase

adulto, sendo identificada com uma etiqueta do modelo B. A constituição dos casais foi registrada na planilha eletrônica, na aba “casais”, seguindo o seguinte procedimento: inserção da data, leitura do código da nova gaiola de casais, inserção dos códigos das gaiolas utilizadas na fase ninfal e nova verificação do sexo dos indivíduos (Figura 6).

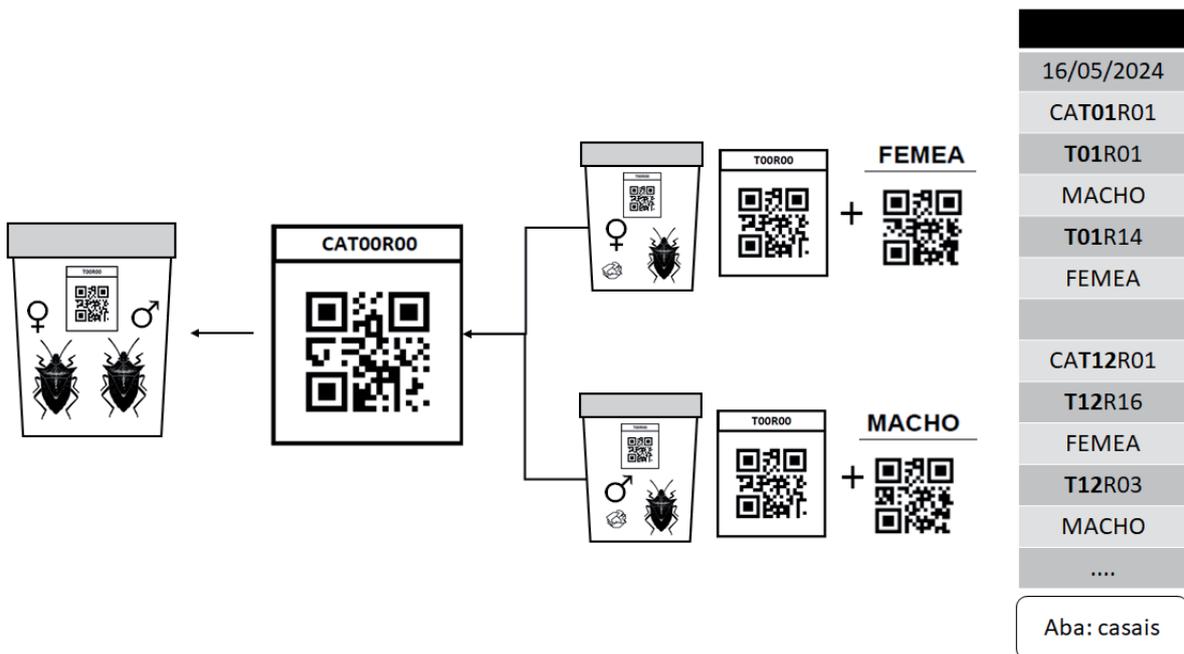


Figura 6. Esquema da organização de casais de *Euschistus heros* (Hemiptera: Pentatomidae) para acompanhamento da fase adulta.

Coleta dos dados biológicos na fase adulta

Diariamente, as gaiolas com os casais foram avaliadas quanto à mortalidade dos adultos e à presença de posturas. A avaliação foi realizada da seguinte forma: abrir-se a aba “posturas” na planilha eletrônica, inseriu-se a data e leu-se o código da gaiola, o que possibilitava duas situações: Presença de adultos mortos – Após a leitura do código da gaiola, registrava-se o sexo do inseto morto (ou de ambos); Presença de posturas – As posturas

foram removidas das gaiolas e acondicionadas individualmente em tubos de microcentrifuga (1,5 mL), previamente etiquetados (Figura 7). A tampa dos tubos foi perfurada e coberta com tecido voil ou com parafilme perfurado, permitindo trocas gasosas e manutenção da umidade. Os tubos foram armazenados em racks apropriados e mantidos dentro de caixas plásticas contendo algodão umedecido para preservar as condições ambientais.

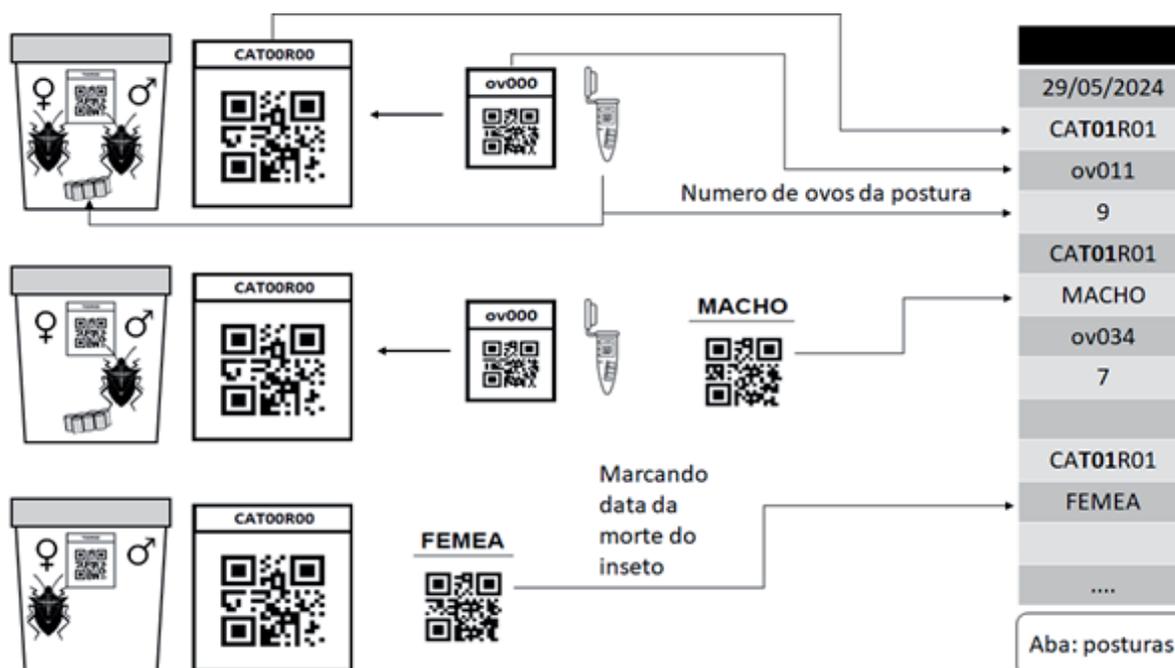


Figura 7. Esquema para coleta da data de morte dos adultos para cálculo de longevidade e rastreamento das posturas de cada tratamento do percevejo *Euschistus heros* (Hemiptera: Pentatomidae)

Avaliação da eclosão dos ovos

Após a eclosão dos ovos, as posturas foram levadas para uma lupa estereoscópica para contagem do número de ovos e de ninfas emergidas de cada uma. A etiqueta de cada tubo era lida, e os dados referentes ao número de ovos e à emergência das ninfas eram inseridos manualmente (Figura 8). Ainda não foi possível automatizar o processo de contagem dos ovos e ninfas, mas as posturas podem ser contabilizadas e armazenadas novamente nos respectivos tubos para posterior conferência.

Organização dos dados biológicos para análise utilizando algoritmos em Python

Para organizar os dados em *data frames* estruturados, foram desenvolvidos três algoritmos (Anexo A) utilizando a linguagem de programação Python, versão 3.7.6, com o uso das bibliotecas *os*, *re* (Python Software Foundation, 2020) e *Pandas*, versão 1.0.1 (The Pandas Development Team,

2020). Parte da lógica e do código dos algoritmos foi desenvolvida com o suporte da ferramenta de inteligência artificial ChatGPT (OpenAI, 2025), que forneceu sugestões com base nas instruções fornecidas.

Os três algoritmos desenvolvidos são: Algoritmo 1 (Anexo A, linhas 11 a 112): organiza os dados da fase ninfal, incluindo trocas de instares, mortalidade e peso no primeiro dia da fase adulta; Algoritmo 2 (Anexo A, linhas 113 a 274): organiza os dados relativos à rastreabilidade das posturas e à emergência dos ovos dos casais; Algoritmo 3 (Anexo A, linhas 245 a 361): calcula a longevidade dos adultos que compõem os casais e realiza a verificação da integridade dos dados. Os algoritmos foram documentados para facilitar futuras melhorias e adaptações a outras situações experimentais.

Toda a estrutura de codificação das etiquetas (T00R00, CAT00R00 e ov000), bem como a nomeação das abas nas planilhas (“ninfal”, “adulto”,

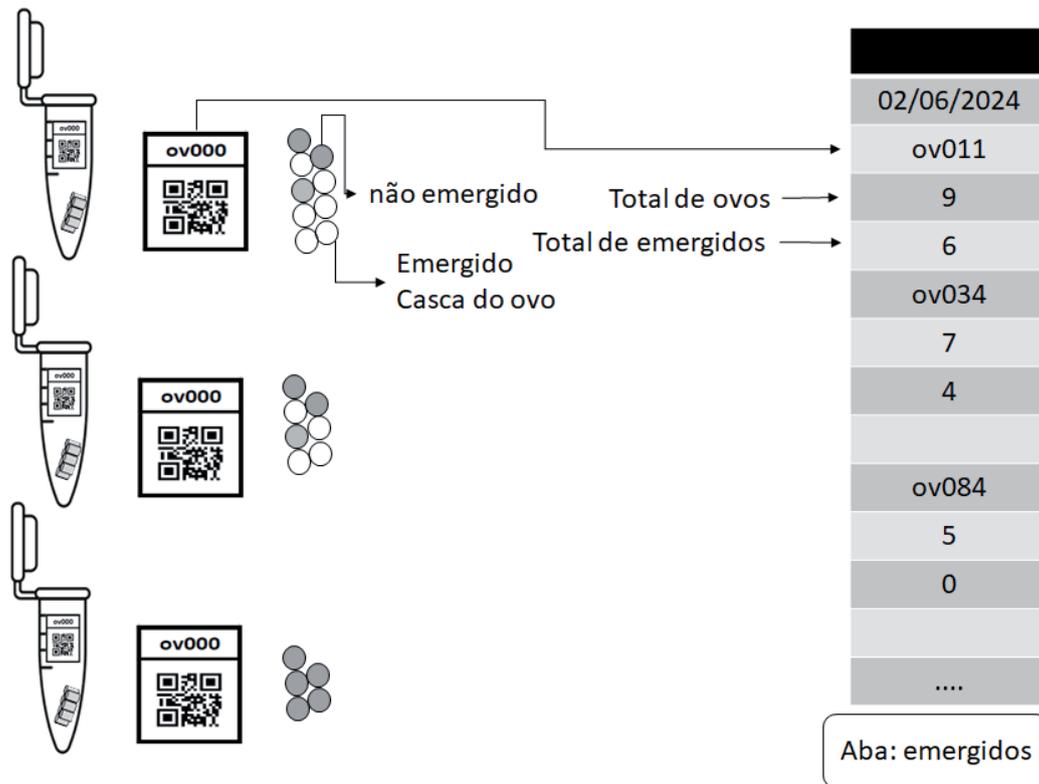


Figura 8. Esquema da avaliação do número de ovos e ninfas emergidas de cada postura de *Euschistus heros* (Hemiptera: Pentatomidae).

“casais”, “posturas” e “emergidos”) e a codificação das variáveis de entrada foram planejadas para garantir a execução dos algoritmos sem a necessidade de ajustes na programação original. Para executar os algoritmos, é necessário apenas informar o diretório de trabalho (Anexo A, linha 7) e o nome do arquivo Excel (Anexo A, linha 8), contendo as abas mencionadas com os dados correspondentes. Como saída, os algoritmos geram quatro arquivos CSV (valores separados por ponto e vírgula), nomeados com o nome original do arquivo Excel como prefixo. Essa estratégia evita sobreposição de arquivos e permite a abertura dos resultados em planilhas eletrônicas para verificação da consistência dos dados ou para análise estatística posterior.

Os algoritmos foram implementados e testados em ambiente interativo, utilizando o Jupyter Notebook (Project Jupyter, 2025).

Análise estatística

Os dados organizados em data frames pelos algoritmos foram tratados, e inconsistências foram investigadas. O delineamento estatístico foi inteiramente casualizado, os dados de contagem foram transformados pelo método de Box-Cox (Sakia, 1992). Os dados de porcentagem (% de emergência) foram transformados pela raiz

arco-seno de x . Devido ao desbalanceamento das repetições entre os tratamentos a análise de variância foi realizada por meio de ajuste por modelos lineares generalizados (GLM), com as médias comparadas pelo teste de Tukey ($p < 0,05$), utilizando-se a interface (IDE) RStudio v. 2024.12.1+563 e a linguagem de programação estatística R v. 4.5.0 para Windows (Rstudio Team, 2025; R Core Team, 2025).

Resultados e Discussão

Foram testadas várias dimensões de etiquetas, considerando o tamanho das gaiolas de bioensaio e dos tubos de microcentrifuga, bem como a necessidade de conter as informações mínimas para permitir o rastreamento dos insetos submetidos aos diferentes tratamentos, tanto na fase ninfal quanto na fase adulta e nas posturas. Assim, as dimensões de 17 x 15 mm para as gaiolas de bioensaio e de 6 x 5 mm para os tubos de microcentrifuga foram escolhidas por permitirem a geração de QR Codes com resolução suficiente para os leitores testados neste experimento, além de possibilitarem a fixação das etiquetas nos respectivos frascos de contenção sem maiores dificuldades ou perda de informação.

O QR Code foi criado em 1994 pela empresa Denso Wave para o rastreamento de peças

automotivas, oferecendo vantagens significativas em relação aos códigos de barras tradicionais. Entre essas vantagens, destacam-se: maior capacidade de armazenamento de dados, podendo conter até 7.089 caracteres numéricos ou 4.296 caracteres alfanuméricos, em comparação aos códigos de barras convencionais; leitura em diferentes ângulos, sem a necessidade de alinhamento do leitor; possibilidade de impressão em tamanhos muito menores, sem perda de legibilidade; e tolerância a danos, uma vez que, mesmo com parte da imagem comprometida, os códigos de correção de erros garantem a leitura bem-sucedida (Denso Wave, 2025). Os códigos alfanuméricos utilizados neste experimento podem ser facilmente adaptados a outras situações e necessidades experimentais, pois a API empregada extrai a informação contida em uma célula da planilha e a converte imediatamente em um QR Code — bastando apenas alterar o conteúdo da célula, sem a necessidade de conhecimento em programação.

Ambos os leitores 2D testados apresentaram vantagens e limitações, mas atenderam às necessidades de leitura das etiquetas elaboradas. O modelo de leitor via Bluetooth (sem fio) apresentou maior dificuldade na leitura dos QR Codes aplicados em etiquetas de ovos revestidas com fita adesiva, provavelmente em razão da resolução do sensor de leitura (640 x 480 pixels), e ao menor tamanho e maior refletividade dessas etiquetas. Por outro lado, o leitor 2D sem fio facilitou a leitura das etiquetas nas gaiolas, contribuindo para a organização da bancada, além de permitir o afastamento de até 10 metros do equipamento de armazenamento (tablet e/ou computador). No entanto, seu custo é mais elevado em comparação aos leitores com fio, além de exigir atenção quanto à carga da bateria. O leitor com fio, por sua vez, apresentou maior rapidez na leitura dos QR Codes, mas a conexão via USB Tipo A limitou seu uso a computadores de mesa ou notebooks compatíveis. No contexto do laboratório, a conexão com um notebook atendeu adequadamente às demandas de mobilidade durante a coleta de dados, sem intercorrências.

A Figura 4 apresenta o processo de inserção de dados referentes à fase ninfal. A não necessidade de utilizar planilhas estruturadas, em que cada informação precisa ser inserida em colunas e linhas específicas, agilizou todo o processo de coleta. Todos os dados são registrados em uma única coluna, e apenas as gaiolas que apresentam alterações — como presença de exúvias, insetos mortos ou adultos emergidos — são anotadas. Esse

formato reduz o tempo de avaliação e minimiza as chances de erros na inserção dos dados.

A conexão da balança analítica com a planilha eletrônica foi simples, sendo que o fabricante disponibilizou as instruções e os drivers de comunicação em seu website, o que facilitou a montagem do sistema de pesagem. Os dados provenientes dos dois equipamentos (balança e leitor) foram coletados sem apresentar incompatibilidades. As informações são armazenadas em uma estrutura unicolunar, o que dispensa o operador de seguir uma sequência predeterminada e reduz a preocupação com possíveis erros na reorganização dos dados. A transcrição manual de dados de pesagem em uma balança analítica exige atenção especial do operador, sendo comum a ocorrência de erros devido ao grande volume de informações que precisam ser anotadas e posteriormente digitadas. Com a adoção deste sistema, todos esses erros foram eliminados.

O processo de conexão de um tubo de microcentrífuga etiquetado, contendo uma postura proveniente de um casal específico, permite a rastreabilidade dessa postura, incluindo informações como data da oviposição, número de ovos e quantidade de ninfas emergidas. Como a leitura dos códigos da gaiola e do tubo é automatizada, não foram observados problemas relacionados a posturas “duplicadas” com o mesmo código, atribuídas a datas ou casais diferentes, ou ainda posturas “desaparecidas”. A possibilidade de realizar a contagem dos ovos e da emergência das ninfas posteriormente permite que as atividades sejam organizadas de acordo com a disponibilidade de tempo para a avaliação do bioensaio.

Todos os dados coletados foram armazenados em planilhas eletrônicas organizadas em abas unicolunares, ou seja, todos os dados foram inseridos em linhas sequenciais de uma única coluna, sem distinção prévia quanto à sua posição ou significado (Figuras 5 a 8). Os três algoritmos em linguagem Python para reorganizar as informações em data frames (Tabelas 1 a 4) funcionaram como esperado. Apesar de ser possível realizar todo o processamento um único algoritmo, optou-se por subdividir em três para facilitar a depuração dos erros de programação e a adaptação dos algoritmos para outras necessidades experimentais. A escolha dessa linguagem de programação se deu por ser uma linguagem interpretada, de alto nível, e por apresentar uma sintaxe simples, o que facilita a leitura, interpretação. Além disso, por ser uma linguagem de código aberto, não há necessidade de aquisição de licenças (Python Software Foundation, 2020).

A execução dos algoritmos gerou quatro data frames, nomeados com o prefixo do nome original do arquivo Excel, seguido pelos sufixos “fase_ninfal” (Tabela 1), “post_emerg_RAW” (Tabela 2), “post_emerg_RES” (Tabela 3) e “fase_adult” (Tabela 4). Os dados foram organizados de forma a facilitar a identificação de inconsistências na entrada das informações, e a gravação em formato CSV permite a importação para qualquer planilha eletrônica, possibilitando a verificação e o preparo para as análises estatísticas.

Na Tabela 1 - Data frame “fase_ninfal”, as datas abaixo das colunas “Instar_NUMERO” indicam o início de cada fase ninfal. O período de cada instar pode ser calculado pela diferença entre as datas. Por exemplo, para o inseto T3-R40 (linha 1), o 3º instar durou cinco dias, de 17/04/2024 até 22/04/2024, quando se iniciou o 4º instar. Nas linhas 6 e 9, nas colunas “Ocorrencia” e “DataOcorr”, estão indicadas a morte ou a perda do inseto. A presença da coluna

“6_instar” aponta uma inconsistência nos dados, pois para percevejos só são possíveis cinco instares. Neste caso específico, o que ocorreu foi uma entrada dupla de datas para uma repetição, o que pode ser facilmente investigado e corrigido ao analisar a totalidade do data frame. No caso do peso de adultos, podem ser observadas outras inconsistências. Para o inseto T3-R05 (linha 5), a coluna “Dataemerg” indica o dia de emergência do adulto, e a coluna “Peso” contém os dados provenientes da balança para este inseto. Ao comparar com os outros dados adjacentes, observa-se que há discrepâncias, o que indica que a coleta de dados desse inseto específico apresentava problemas, outro ponto é que este inseto estava morto ao ser pesado, o que é indicado na coluna “Estado”. A forma como os dados foram organizados permite que essas inconsistências sejam detectadas, e medidas de correção ou eliminação dos dados discrepantes podem ser implementadas.

Tabela 1. Data frame “fase_ninfal” gerado pelo Algoritmo 1 a partir da organização dos dados das abas “ninfal” e “adulto” um arquivo de planilha eletrônica.

id	Trat_AbN	Ocorrencia	Data_Ocorr	Instar_3	Instar_4	Instar_5	Instar_6
1	T3-R40			17/04/2024	22/04/2024	28/04/2024	
2	T1-R40			17/04/2024	22/04/2024	28/04/2024	
3	T3-R01			18/04/2024	23/04/2024	29/04/2024	
4	T3-R03			18/04/2024	22/04/2024	28/04/2024	
5	T3-R05			18/04/2024	24/04/2024	02/05/2024	
6	T2-R07	MORTO	18/04/2024	18/04/2024			
7	T2-R06	MORTO	18/04/2024	18/04/2024			
8	T1-R08			18/04/2024	23/04/2024	28/04/2024	
9	T3-R10	PERDIDO	18/04/2024	18/04/2024			
10	T1-R09			18/04/2024	22/04/2024	26/04/2024	
...

id	Trat_AbN	Dataemerg	Sexo_AbN	Peso	Estado
1	T3-R40	06/05/2024	FEMEA	0.0531	
2	T1-R40				
3	T3-R01	07/05/2024	FEMEA	0.0567	
4	T3-R03	06/05/2024	MACHO	0.0721	
5	T3-R05	13/05/2024	MACHO	0.429	MORTO
6	T2-R07				
7	T2-R06				
8	T1-R08	06/05/2024	FEMEA	0.0735	
9	T3-R10				
10	T1-R09	02/05/2024	FEMEA	0.0817	
...

Nota: id = identificador da amostra; Trat_AbN = identificação do tratamento e repetição proveniente da aba “Ninfal” do arquivo Excel; Ocorrencia = indica morte ou perda na fase ninfal; Data_Ocorr = data do evento Ocorrencia; Instar_NUMERO = indica a data da troca de cada instar; Dataemerg = data da emergência do adulto; Sexo_AbN = identificação do sexo da amostra proveniente da aba “Ninfal”; Peso = massa do adulto no primeiro dia em grama; Estado = indica que o inseto atingiu a fase adulta e morreu.

O data frame “post_emerg_RAW” (Tabela 2) apresenta os dados brutos reorganizados, mas sem deleção, de cada postura e sua respectiva associação aos casais. Assim, é possível verificar, por exemplo, que o casal CAT2-R01 (linhas 1, 3 e 9) apresentou posturas nos dias 07, 09 e 20/05/2024. Como exemplo de inconsistência, observa-se um erro na linha 1, em que o número de ninfas emergidas (n=20) é maior que o número de ovos da postura (n=13). Como esses dados são digitados

manualmente, houve uma inversão na entrada das informações. O erro foi corrigido ao reavaliar o tubo ov001. Várias informações biológicas derivadas podem ser extraídas deste data frame, como o período de pré-oviposição, o número total de posturas, a média de ovos por postura, o período de oviposição e a taxa de emergência. Essas informações consolidadas foram organizadas no data frame “post_emerg_RES” (Tabela 3).

Tabela 2. Data frame “post_emerg_RAW” gerado pelo Algoritmo 2 a partir da organização dos dados das abas “postura” e “emergido” da planilha eletrônica.

id	Datapostura	Casal_AbP	Postura	ndeovos	DataLeitura	ndeemerg
1	07/05/2024	CAT2-R01	ov001	13	26/06/2024	20.0
2	07/05/2024	CAT2-R02	ov002	5	04/07/2024	0.0
3	09/05/2024	CAT2-R01	ov048	8	04/07/2024	0.0
4	09/05/2024	CAT2-R02	ov072	13	04/07/2024	0.0
5	09/05/2024	CAT2-R03	ov073	17	04/07/2024	2.0
6	09/05/2024	CAT2-R05	ov046	14	04/07/2024	8.0
7	09/05/2024	CAT2-R04	ov025	28	04/07/2024	0.0
8	09/05/2024	CAT3-R05	ov062	7	05/07/2024	0.0
9	20/05/2024	CAT2-R01	ov062	7	05/07/2024	0.0
10	09/05/2024	CAT3-R02	ov019	9	04/07/2024	0.0
...

Nota: id = identificador da amostra; Datapostura = data da oviposição; Casal_AbP = identificação casal do tratamento e repetição proveniente da aba “Postura” do arquivo Excel; Postura = identificação da postura; ndeovos= número de ovos da postura; DataLeitura = data que foi avaliada a emergência das ninfas; ndeemerg = número de ninfas emergidas.

Tabela 3. Data frame “post_emerg_RES” gerado pelo Algoritmo 2 a partir da organização e resumo dos dados das abas “postura” e “emergido” da planilha eletrônica, apresentando a totalização das posturas, ovos e ninfas emergidas por fêmeas de cada casal.

id	Casal_AbP	inicio_da_ov	final_de_ov	totovos	totpost	totemerg	Sexo
1	CAT1-R01	10/05/2024	03/06/2024	69.0	10.0	3.0	FEMEA
2	CAT1-R02	10/05/2024	10/05/2024	14.0	1.0	11.0	FEMEA
3	CAT1-R03	15/05/2024	21/05/2024	38.0	4.0	45.0	FEMEA
4	CAT1-R04	15/05/2024	17/05/2024	17.0	2.0	7.0	FEMEA
5	CAT1-R05	10/05/2024	07/06/2024	71.0	11.0	0.0	FEMEA
6	CAT1-R06	10/05/2024	21/05/2024	72.0	7.0	0.0	FEMEA
7	CAT1-R12	15/05/2024	07/06/2024	90.0	6.0	0.0	FEMEA
8	CAT1-R20	22/05/2024	27/06/2024	52.0	5.0	4.0	FEMEA
9	CAT2-R1	07/05/2024	20/05/2024	59.0	5.0	29.0	FEMEA
10	CAT2-R12	03/06/2024	26/06/2024	190.0	13.0	93.0	FEMEA
...

Nota: id = identificador da amostra, Casal_AbP = identificação casal do tratamento e repetição proveniente da aba “Postura” do arquivo Excel, inicio_da_ov = data da primeira postura, final_de_ov = data da última postura, totovos = somatória de todos os ovos depositados pela fêmea, totpost = somatória de todas as posturas depositadas pela fêmea, totemerg = somatória de todas as ninfas emergidas das posturas depositadas pela fêmea, Sexo = sexo para verificação da consistência das informações do casal, esta coluna garante que uma linha para cada sexo de cada casal seja criada, permite verificar se alguma fêmea não ovipositou, ou se a morte de um macho não foi registrada.

O data frame “fase_adult” (Tabela 4) permite calcular a longevidade dos adultos utilizados para a formação de casais, por meio da subtração entre a data de emergência (“Dataemerg”) e a data de morte (“Datamorte”). As colunas redundantes, como

“Casal_AbC”, “Casal_AbP”, “Sexo_AbC”, “Sexo_AbN” e “Sexo_AbP” - oriundas de diferentes abas de planilha - possibilitam verificar se a fusão dos data frames foi realizada de forma correta e rastrear dados inconsistentes até a sua origem.

Tabela 4. Data frame “fase_adult” gerado pelo Algoritmo 3 a partir da organização dos dados da aba “Casais” um arquivo de planilha eletrônica, e fusão dos data frames “fase_ninfa” e “post_emerg_RES”.

Casal_AbC	Datacasal	Trat_AbC	Sexo_AbC	Trat_AbN	Dataemerg	Datamorte	Sexo_AbN	Peso
CAT1-R01	03/05/2024	T1-R11	FEMEA	T1-R11	02/05/2024	10/06/2024	FEMEA	0.0766
CAT1-R01	03/05/2024	T1-R39	MACHO	T1-R39	02/05/2024	13/05/2024	MACHO	0.0739
CAT1-R02	03/05/2024	T1-R02	FEMEA	T1-R02	02/05/2024	13/05/2024	FEMEA	0.0671
CAT1-R02	03/05/2024	T1-R07	MACHO	T1-R07	02/05/2024	03/06/2024	MACHO	0.0725
CAT1-R03	03/05/2024	T1-R09	FEMEA	T1-R09	02/05/2024	22/05/2024	FEMEA	0.0817
CAT1-R03	03/05/2024	T1-R32	MACHO	T1-R32	02/05/2024	27/05/2024	MACHO	0.0779
CAT2-R01	03/05/2024	T2-R35	MACHO	T2-R35	02/05/2024	27/05/2024	MACHO	0.0625
CAT2-R01	03/05/2024	T2-R05	FEMEA	T2-R05	02/05/2024	28/05/2024	FEMEA	0.0879
CAT2-R02	03/05/2024	T2-R27	MACHO	T2-R27	02/05/2024	07/06/2024	MACHO	0.0744
...

Casal_AbP_x	Sexo_AbP	inicio_da_ov	final_de_ov	totpost	totovos	totemerg	Sexo
CAT1-R01	FEMEA	10/05/2024	03/06/2024	10.0	69.0	3.0	FEMEA
CAT1-R01	MACHO						MACHO
CAT1-R02	FEMEA	10/05/2024	10/05/2024	1.0	14.0	11.0	FEMEA
CAT1-R02	MACHO						MACHO
CAT1-R03	FEMEA	15/05/2024	21/05/2024	4.0	38.0	45.0	FEMEA
CAT1-R03	MACHO						MACHO
CAT2-R1	MACHO						MACHO
CAT2-R1	FEMEA	07/05/2024	20/05/2024	5.0	59.0	29.0	FEMEA
CAT2-R2	MACHO						MACHO
...

Nota: id = identificador da amostra, AbN = dados proveniente da aba “Ninfa” do arquivo Excel, AbA = dados proveniente da aba “Adulto”, AbC = dados proveniente da aba “Casais”, AbP = dados proveniente da aba “Posturas”, AbE = dados proveniente da aba “Emergidos”, Datacasal = data da formação do casal, Casal_AbP = identificação do casal, Trat_AbC = inseto original usado para formar o casal, Sexo_AbC = identificação do sexo da inseto, Trat_AbN = identificação do inseto original, Dataemerg (AbN) = data da emergência do adulto, Datamorte (AbP) = data da morte do adulto, Peso (AbA) = massa do adulto no primeiro dia em grama, inicio_da_ov (AbP) - data da primeira postura, final_de_ov (AbP) - data da última postura, totpost (AbP) - total de posturas de cada fêmea, totovos (AbP) - total de ovos de cada fêmea, totemerg (AbE) - total de ninfas emergidas das posturas.

Após o tratamento dos dados brutos, os *data frames* resultantes podem ser facilmente reorganizados para o formato exigido pelo Sistema de Informação de Experimentos da Embrapa (SIExp-Embrapa, 2025). Esse sistema é responsável por estruturar e padronizar os dados experimentais, tornando-os inteligíveis e acessíveis, de acordo com as normas institucionais. Assim, os dados reorganizados podem então ser carregados e armazenados no sistema. Apesar do SIExp contar com funcionalidades para a utilização de coletores de dados, optou-se por uma opção de mais fácil

customização, mas com a garantia que os dados fossem armazenados de acordo com as regras da Empresa.

O tempo de desenvolvimento em cada instar foi similar ao observado em outros estudos biológicos realizados com *E. heros*. Nos três tratamentos avaliados, a sobrevivência na fase ninfa foi superior a 70% (Tabela 5). Em comparação, a maior taxa de sobrevivência obtida por Fortes et al. (2006) foi de 81,6%, utilizando uma dieta natural composta por sementes de amendoim, sementes de soja e frutos de ligustro (*Ligustrum lucidum* Ait., Oleaceae).

Tabela 5. Tempo de desenvolvimento (dias) em cada instar, razão sexual [fêmeas/ (fêmeas + machos)], e sobrevivência total (%) de *Euschistus heros* (Hemiptera: Pentatomidae) alimentados com sementes de soja e amendoim, mantidos em sala de criação (26 ± 2 °C, fotofase de 14 horas).

Tratamento	Tempo de desenvolvimento (dias)				Razão sexual	Sobrevivência (%)
	(n)					
	2° instar	3° instar	4° instar	5° instar		
Soja (SO)	6,8±0,59a (39)	4,8±0,07 b (38)	5,0±0,44a (33)	7,6±0,32a (28)	0,46	70
Amendoim (AM)	6,4±0,35a (38)	5,4±0,08 a (36)	4,3±0,36a (31)	7,2±0,24a (30)	0,58	75
SO + AM	6,3±0,16a (35)	5,2±0,17ab (34)	5,0±0,47a (33)	7,7±0,32a (28)	0,57	70
CV (%)	23,33	15,57	33,01	16,71		

* Médias seguidas pela mesma letra na coluna não diferem significativamente pelo teste de Tukey ($p < 0,05$).

A massa dos adultos (mg) de *E. heros* no primeiro dia de vida adulta corrobora com outros estudos sobre a biologia deste percevejo. Não houve diferenças estatísticas significativas entre os tratamentos; mas, entre os sexos, as fêmeas foram significativamente mais pesadas e se desenvolveram mais rapidamente que os machos (Figura 9). O peso dos adultos foi semelhante ao obtido por Mendoza et al. (2016), utilizando dietas artificiais, e por Oliveira et al (2016), com dietas naturais, nas quais as fêmeas foram mais pesadas que os machos. Esse resultado é esperado, pois as fêmeas de insetos são normalmente maiores que os machos (Stillwell et al., 2010).

O tempo de desenvolvimento observado em nosso experimento foi de 22,9 dias para fêmeas e 24,6 dias para machos, sob temperatura de 26 ± 2 °C (Figura 9). O menor tempo de desenvolvimento das

fêmeas é um dado inesperado, pois, segundo Teder (2014), em um estudo de metanálise de biologia de insetos, concluiu-se que, na maioria das espécies (79%), o sexo de maior tamanho também apresenta um tempo de desenvolvimento larval mais longo. Mendoza et al. (2016), testando sete formulações de dietas artificiais, obtiveram um tempo médio de desenvolvimento de 27,7 dias (do segundo instar ao adulto, a 25 ± 2 °C). Fortes et al. (2006) obtiveram um tempo de desenvolvimento de $24,1 \pm 0,16$ dias utilizando uma dieta natural (sementes de amendoim, sementes de soja e frutos de ligustro) a 25 ± 1 °C, mas ambos os experimentos não foram feitos a distinção do tempo de desenvolvimento entre os sexos. É possível que o menor tempo de desenvolvimento observado em nosso experimento seja resultado da temperatura média mais elevada da sala de criação.

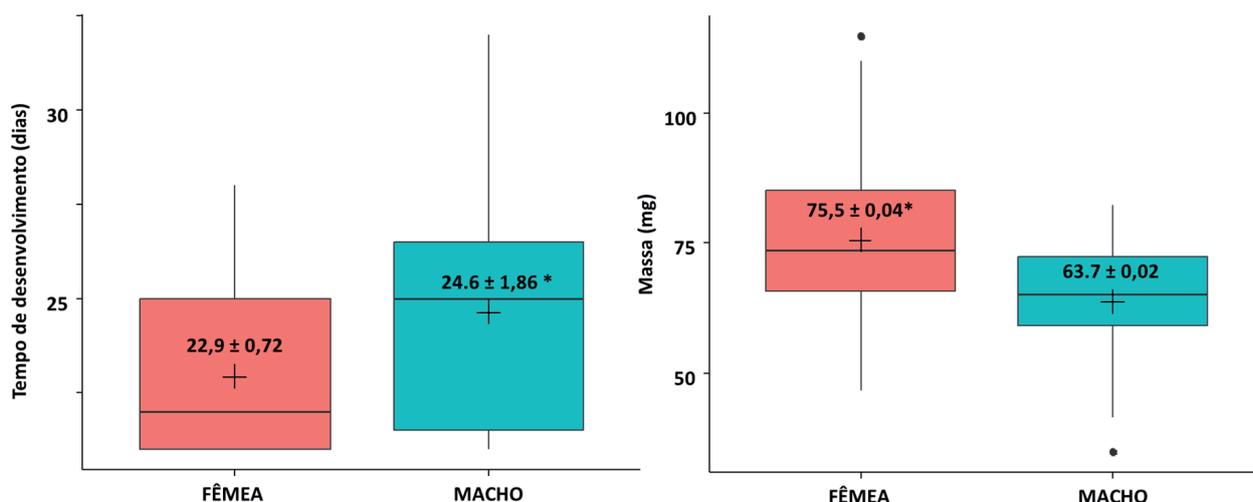


Figura 9. Boxplots do tempo de desenvolvimento (dias) (2° instar - adulto) e massa (mg) ao primeiro dia do adulto de *Euschistus heros* (Hemiptera: Pentatomidae) alimentados com sementes de soja e amendoim mantidos em sala de criação (26 ± 2 °C, fotofase de 14 horas).

Quanto à longevidade dos adultos, não foram observadas diferenças significativas entre os sexos, mas foram encontradas diferenças entre os tratamentos alimentares na fase adulta: os insetos alimentados exclusivamente com sementes de soja apresentaram menor longevidade (24,2 dias) em comparação àqueles alimentados sementes de amendoim (48,2 dias) ou com a combinação de soja + amendoim (49,9 dias). Essa maior longevidade repercutiu no número de posturas e, conseqüentemente, no número de ovos, que também diferiram significativamente entre os tratamentos (Tabela 6). Costa et al. (1998), conduzindo um experimento em laboratório, encontraram resultados de longevidade média semelhantes, com 51,2 dias para fêmeas e 46,5 dias para machos.

Em relação ao período pré-oviposição, este variou entre 10,2 e 12,4 dias entre os tratamentos, sem diferenças estatísticas (Tabela 6), o que é

compatível com os resultados obtidos por Costa et al. (1998). A longevidade, número de posturas e ovos foram significativamente maiores para as dietas com amendoim e soja + amendoim. Esse resultado é esperado devido a interdependência das variáveis, onde fêmeas mais longevas têm maior oportunidade de depositar posturas e ovos (Tabela 6). Neste experimento, foram rastreadas 276 posturas, totalizando 3307 ovos. Estudos adicionais sobre a adequação desses alimentos ainda serão necessários para avaliar se os parâmetros de longevidade e fecundidade são decorrentes da alimentação durante a fase ninfal ou apenas durante a fase adulta. No entanto, este experimento confirma que o uso de sementes de amendoim como complemento na dieta alimentar nos sistemas de criação de percevejo pragas da soja é importante e recomendado.

Tabela 6. Longevidade e período de pré-oviposição (dias), número total de posturas e total de ovos (média ± Erro Padrão) de *Euschistus heros* (Hemiptera: Pentatomidae) alimentados com sementes de soja e amendoim e mantidos em sala de criação (26 ± 2 °C, fotofase de 14 horas).

Tratamentos	Longevidade (dias)	Pre-oviposição (dias)	Total de posturas (n)	Total de ovos (n)
Soja (So)	24,2 ± 59,71 b	10,3 ± 3,81 a	5,8 ± 4,42 b	52,9 ± 270,36 b
Amendoim (Am)	48,2 ± 168,26 a	11,9 ± 28,77 a	11,6 ± 21,10 a	170,1 ± 3889,07 a
So + Am	49,9 ± 175,89 a	12,4 ± 16,74 a	11,4 ± 7,11 a	118,3 ± 2045,65 a

* Médias seguidas pela mesma letra na coluna não diferem significativamente pelo teste de Tukey (p<0,05).

** Os dados foram transformados pelo método de Box-Cox (SAKIA, 1992) para a realização da análise de variância.

Com relação a taxa de emergência das ninfas, não houve diferenças estatísticas entre os tratamentos, variando de 24,7% (T1 - alimentados com soja) a 45,6% (T2 - alimentados com amendoim). Essa baixa taxa de emergência pode ser decorrente de deficiências nutricionais nas dietas. Novos estudos serão necessários para confirmar estes resultados.

Conclusão

Este trabalho é um exemplo prático de como é possível estabelecer protocolos de coleta dos dados de forma automatizada, integrando tecnologias em um laboratório de entomologia aplicada, gerando ganhos de eficiência e facilitando todo o processo de análise dos resultados.

Referências

BARCHARD, K. A.; PACE, L. A. Preventing human error: The impact of data entry methods on data accuracy and statistical results. **Computers in Human Behavior**, v. 27, n. 5, p. 1834-1839, 2011. DOI: 10.1016/j.chb.2011.04.004.

COSTA, M. L. M.; BORGES, M.; VILELA, E. F. Biologia reprodutiva de *Euschistus heros* (F.) (Heteroptera: Pentatomidae). **Anais da Sociedade Entomológica do Brasil**, v. 27, n. 5, p. 559-568, 1998. DOI: 10.1590/S0301-80591998000400008.

DENSO WAVE. **QR Code development story**. [2025]. Disponível em: <https://www.denso-wave.com/en/technology/vol1.html>. Acesso em: 16 abr. 2025.

FORTES, P.; MAGRO, S. R.; PANIZZI, A. R.; PARRA, J. R. P. Development of a dry artificial diet for *Nezara viridula* (L.) and *Euschistus heros* (Fabricius) (Heteroptera: Pentatomidae). **Neotropical Entomology**, v. 35, n. 5, p. 567-572, 2006. DOI: 10.1590/S1519-566X2006000500001.

GOQR.ME. **QR Code Generator**. [2025]. Disponível em: <https://goqr.me/api/>. Acesso em: 16 abr. 2025.

KANZA, S.; WILLOUGHBY, C.; GIBBINS, N. WHITBY, R.; FREY, J. G.; ERJAVEC, J.; ZUPANCIC, K.; HREN, K.; KOVAC, K. Electronic lab notebooks: can they replace paper? **Journal of Cheminformatics**, v. 9, n. 1, p. 31, 2017. DOI: 10.1186/s13321-017-0221-3.

LEY, B.; RIJAL, K. R.; MARFURT, J.; ADHIKARI, N. R.; BANJARA, M. R.; SHRESTHA, U. T.; THRIEMER, K.; PRICE, R. N.; GHIMIRE, P. Analysis of erroneous data entries in paper based and electronic data collection. **BMC Research Notes**, v. 12, article 537, 2019. DOI: 10.1186/s13104-019-4574-8.

MENDOZA, A. C.; ROCHA, A. C. P. da; PARRA, J. R. P. Lyophilized artificial diet for rearing the Neotropical *Euschistus heros* (Hemiptera: Pentatomidae). **Journal of Insect Science**, v. 16, n. 1, 41, 2016. 9 p. DOI: 10.1093/jisesa/iew025.

OLIVEIRA, D. G. P.; DUDCZAK, A. C.; ALVES, L. F. A.; SOSA-GÓMEZ, D. R. Biological parameters of *Euschistus heros* (F.) (Heteroptera: Pentatomidae) and its susceptibility to entomopathogenic fungi when fed on different diets. **Brazilian Archives of Biology and Technology**, v. 59, e16150141, 2016. DOI: 10.1590/1678-4324-2016150141.

OPENAI. **ChatGPT**: versão GPT-4. [S.I.]: OpenAI, 2025. Disponível em: <https://chat.openai.com/>. Acesso em: 16 abr. 2025.

PANIZZI, A. R.; LUCINI, T. Host plant-stinkbug (Pentatomidae) relationships. In: COKL, A.; BORGES, M. (ed.). **Stinkbugs**: biorational control based on communication processes. Boca Raton: CRC Press, 2017. p. 30-57.

PEREIRA, F. G. F.; ATAÍDE, M. B. C.; SILVA, R. L.; NÉRI, E. D. R.; CARVALHO, G. C. N.; CAETANO, J. A. Environmental variables and errors in the preparation and administration of medicines. **Revista Brasileira de Enfermagem**, v. 71, n. 3, p. 1046-1054, 2018. DOI: 10.1590/0034-7167-2016-0041.

POSSEBOM, T.; LUCINI, T.; PANIZZI, A. R. Stink bugs nymph and adult biology and adult preference on cultivated crop plants in the Southern Brazilian neotropics. **Environmental Entomology**, v. 49, n. 1, p. 132-140, 2020. DOI: 10.1093/ee/nvz142.

PROJECT JUPYTER. **Jupyter Notebook**: The Classic Notebook Interface. San Francisco, 2025. Disponível em: <https://jupyter.org/>. Acesso em: 16 abr. 2025.

PYTHON SOFTWARE FOUNDATION. **Python**: linguagem de programação (versão 3.7.6). [S.I.]: [S. I.], 2020.

Disponível em: <https://www.python.org/>. Acesso em: 16 abr. 2025.

R CORE TEAM. **R**: A language and environment for statistical computing. Version 4.5.0. Vienna: R Foundation for Statistical Computing, 2025. Disponível em: <https://www.r-project.org/>. Acesso em: 22 abr. 2025.

RSTUDIO TEAM. **RStudio**: Integrated Development for R. Boston, MA: RStudio, PBC, 2025. Disponível em: <https://posit.co/>. Acesso em: 22 abr. 2025.

SAKIA, R. M. The Box-Cox Transformation Technique: A Review. **Journal of the Royal Statistical Society, Series D, The Statistician**, v. 41, n. 2, p. 169-178, 1992. DOI: 10.2307/2348250.

SIExp-EMBRAPA. **Dados Experimentais coletados a partir de dispositivos móveis e sincronizados com a nuvem do SIExp**. Brasília, DF: Embrapa, [2025]. Disponível em: <https://www.siexp.cnptia.embrapa.br/siexp-mweb/>. Acesso em: 30 jun. 2025.

SOUTHWOOD, T. R. E. The construction, description and analysis of age-specific life-tables. In: SOUTHWOOD, T. R. E. (ed.). **Ecological methods with particular reference to the study of insect populations**. 2nd ed. London: Chapman and Hall, 1978. cap. 25, p. 365-376.

STILLWELL, R. C.; BLANCKENHORN, W. U.; TEDER, T.; DAVIDOWITZ, G.; FOX, C. W. Sex differences in phenotypic plasticity affect variation in sexual size dimorphism in insects: from physiology to evolution. **Annual Review of Entomology**, v. 55, p. 227-245, 2010. DOI: 10.1146/annurev-ento-112408-085500.

TEDER, T. Sexual size dimorphism requires a corresponding sex difference in development time: a meta-analysis in insects. **Functional Ecology**, v. 28, n. 2, p. 479-486, 2014. DOI: 10.1111/1365-2435.12172.

THE PANDAS DEVELOPMENT TEAM. **Pandas-dev/pandas**: Pandas (versão 1.0.1). [S.I.]: Zenodo, 2020. Disponível em: <https://pandas.pydata.org/>. Acesso em: 16 abr. 2025.

TOMACHESKI, J. F.; PANIZZI, A. R.; LUCINI, T. Nymphal and adult biology of the red-shouldered stink bug, *Thyanta perditor* (F.), on cultivated and on wild plants. **Arthropod-Plant Interactions**, v. 13, n. 1, p. 91-98, 2019. DOI: 10.1007/s11829-018-9627-8.

Anexo A. Algoritmos em linguagem Python versão 3.7.6.

```

1      # Algoritmos para organizar os dados biológicos de percevejo elaborados com o auxílio do
ChatGPT 4.0
2      # CARREGAR PACOTES PARA PROCESSAMENTO DOS ALGORITMOS
3      import pandas as pd
4      import os
5      import re
6      # Definir caminhos do arquivo Excel
7      diretorio = "C:\\Users\\PASTA"
8      arquivo_excel = os.path.join(diretorio, "Biologia1.xlsx")
9      # Obtendo o nome base do arquivo original (sem extensão)
10     nome_base = os.path.splitext(os.path.basename(arquivo_excel))[0]

11     # *** Algoritmo 1 organiza os dados da fase ninfal - troca de instares, mortalidade e peso ao
primeiro dia
12     # Ler a aba "ninfa" e "adulto" do arquivo Excel sem cabeçalhos
13     df_excelnf = pd.read_excel(arquivo_excel, sheet_name="ninfa", header=None, dtype=str)
14     df_excelad = pd.read_excel(arquivo_excel, sheet_name="adulto", header=None, dtype=str)
15     # Verificar e converter apenas os valores que são datas
16     df_excelnf = df_excelnf.applymap(lambda x: pd.to_datetime(x, errors='coerce').strftime('%d/%m/%Y')
if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
17     df_excelad = df_excelad.applymap(lambda x: pd.to_datetime(x, errors='coerce').strftime('%d/%m/%Y')
if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
18     # --- Primeira etapa: Criar df1 ligar uma ocorrencia (morte ou fuga da ninfa) ---
19     df1 = df_excelnf
20     df1 = df1.rename(columns={0: 'Original'})
21     df1["Data"] = df1["Original"].apply(lambda x: x if isinstance(x, str) and "/" in x else None)
22     df1["Data"] = df1["Data"].ffill()
23     df1 = df1[df1["Original"] != df1["Data"]]
24     df1 = df1.rename(columns={"Original": "Tratamento"})
25     df1["Ocorrencia"] = df1["Tratamento"].where(~df1["Tratamento"].str.startswith("T"))
26     df1["Ocorrencia"] = df1["Ocorrencia"].shift(-1)
27     df1["Data_Ocorr"] = df1["Data"].where(df1["Ocorrencia"].notna())
28     df1 = df1[df1["Tratamento"].str.startswith("T")]
29     df1 = df1[["Tratamento", "Ocorrencia", "Data_Ocorr"]].drop_duplicates()
30     df1 = df1[~(df1["Ocorrencia"].isna() & df1["Tratamento"].duplicated(keep=False))]
31     # --- Segunda etapa: Criar df2 com agrupamento das datas apenas para "T*" ---
32     df2 = df_excelnf
33     df2 = df2.rename(columns={0: 'Original'})
34     df2["Data"] = df2["Original"].apply(lambda x: x if isinstance(x, str) and "/" in x else None)
35     df2["Data"] = df2["Data"].ffill()
36     df2 = df2[df2["Original"] != df2["Data"]]
37     df2 = df2.rename(columns={"Original": "Tratamento"})
38     df2 = df2[df2["Tratamento"].str.startswith("T")]
39     df2["Data"] = pd.to_datetime(df2["Data"], format="%d/%m/%Y")
40     df2 = df2.sort_values(by=["Tratamento", "Data"], ascending=[True, True])
41     agrupado = {}
42     for _, row in df2.iterrows():
43         valor = row["Tratamento"]
44         data = row["Data"].strftime("%d/%m/%Y")
45         if valor not in agrupado:
46             agrupado[valor] = [data]
47         else:

```

```
48     agrupado[valor].append(data)
49     df2 = pd.DataFrame.from_dict(agrupado, orient="index")
50     df2.columns = [f"Instar_{i+3}" for i in range(df2.shape[1])]
51     df2 = df2.reset_index().rename(columns={"index": "Tratamento"})
52     # --- Terceira etapa: Combinar df1 e df2 ---
53     df_ninfa = pd.merge(df1, df2, on="Tratamento", how="outer")
54     # --- Processamento aba adulto - organizando dados de peso ao 1 dia ---
55     df3 = df_excelad
56     df3 = df3.rename(columns={0: 'Original'})
57     linhas = df3["Original"].tolist()
58     registros = []
59     data_atual = ""
60     codigo = ""
61     sexo = ""
62     peso = ""
63     estado = ""
64     for linha in linhas:
65         linha = linha.strip()
66         if linha:
67             if "/" in linha:
68                 data_atual = linha
69             elif "T" in linha and "-" in linha:
70                 if codigo and sexo and peso:
71                     registros.append([codigo, data_atual, sexo, float(peso), estado])
72                 codigo = linha
73                 estado = ""
74                 sexo = ""
75                 peso = ""
76             elif linha == "MORTO":
77                 estado = "MORTO"
78             elif linha in ["MACHO", "FEMEA"]:
79                 sexo = linha
80             else:
81                 try:
82                     peso = linha.replace(",", ".")
83                     peso = float(peso)
84                     if codigo and sexo and data_atual:
85                         registros.append([codigo, data_atual, sexo, peso, estado])
86                     codigo = ""
87                     sexo = ""
88                     peso = ""
89                     estado = ""
90             except ValueError:
91                 pass
92     registros = [registro for registro in registros if "T" in registro[0]]
93     df_ad = pd.DataFrame(registros, columns=["Tratamento", "Data", "Sexo_AbN", "Peso", "Estado"])
94     # --- Unindo os dois arquivos pelo campo "Tratamento" ---
95     df_final = pd.merge(df_ninfa, df_ad, on="Tratamento", how="outer")
96     df_final = df_final.rename(columns={"Tratamento": "Trat_AbN", "Data": "Dataemerg"})
97     # Descrição das colunas do data frame "fase_ninfa":
98     # Trat_AbN = identificação do tratamento e repetição proveniente da aba "Ninfas" do arquivo Excel;
99     # Ocorrencia = indica morte ou perda na fase ninfa;
100    # Data_Ocorr = data do evento Ocorrencia;
101    # Instar_NUMERO = indica a data da troca de cada instar;
```

```

102 # Dataemerg = data da emergencia do adulto;
103 # Sexo_AbN = identificação do sexo da amostra proveniente da aba "Ninfas";
104 # Peso = massa do adulto no primeiro dia em grama;
105 # Estado = indica que o inseto atingiu a fase adulta e morreu.
106 # Nome do arquivo final
107 arquivo_final = os.path.join(diretorio, f"{nome_base}_fase_ninfal.csv")
108 df_final.to_csv(arquivo_final, sep=";", index=False, encoding="utf-8-sig")
109 # Visualizar as 5 primeiras linhas do DF e local de gravação do arquivo CSV
110 print("\nFase Ninfal:")
111 print(df_final.head())
112 print(f"\nArquivo salvo como: {arquivo_final}")

113 # *** Algoritmo 2 organiza dados das posturas, eclosão dos ovos e realiza totalização por casal
114 # Ler a aba "ninfa" e "adulto" do Excel sem cabeçalhos
115 df_excelpost = pd.read_excel(arquivo_excel, sheet_name="posturas", header=None, dtype=str)
116 df_excelemerg = pd.read_excel(arquivo_excel, sheet_name="emergidos", header=None, dtype=str)
117 # Verificar e converter apenas os valores que são datas
118 df_excelpost = df_excelpost.applymap(lambda x: pd.to_datetime(x, errors='coerce').
strftime('%d/%m/%Y') if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
119 df_excelemerg = df_excelemerg.applymap(lambda x: pd.to_datetime(x, errors='coerce').
strftime('%d/%m/%Y') if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
120 # Processar aba "postura"
121 df4 = df_excelpost
122 df4 = df4.rename(columns={0: 'Original'})
123 linhas = df4["Original"].tolist()
124 dados = []
125 data_postura = ""
126 casal = ""
127 for linha in linhas:
128     if "/" in linha and len(linha) == 10: # detecta data no formato dd/mm/aaaa
129         data_postura = linha
130     elif linha.startswith("CA"):
131         casal = linha
132     elif linha.startswith("ov"):
133         postura = linha
134         # tenta obter o valor logo após a etiqueta iniciada oc ov, se for um número
135         indice = linhas.index(linha)
136         if indice + 1 < len(linhas) and linhas[indice + 1].isdigit():
137             ndeovos = linhas[indice + 1]
138         else:
139             ndeovos = ""
140         dados.append([data_postura, casal, postura, ndeovos])
141 df_postura = pd.DataFrame(dados, columns=["Datapostura", "Casal_AbP", "Postura", "ndeovos_AbP"])
142 # Processar a aba "emergidos"
143 df5 = df_excelemerg
144 df5 = df5.rename(columns={0: 'Original'})
145 linhas = df5["Original"].tolist()
146 rows = []
147 current_date = None
148 i = 0
149 while i < len(linhas):
150     linha = linhas[i].strip()
151     if linha.count("/") == 2:
152         current_date = linha

```

```
153     elif linha.startswith("ov"):
154         ovo = linha
155         valor1 = int(linhas[i+1].strip()) if i+1 < len(linhas) else None
156         valor2 = int(linhas[i+2].strip()) if i+2 < len(linhas) and linhas[i+2].strip().isdigit() else None
157         rows.append([current_date, ovo, valor1, valor2])
158         i += 1
159 df_emerg = pd.DataFrame(rows, columns=["DataLeitura", "Postura", "ndeemerg", "ndeovos"])
160 # Unir os dois dataframes pelo campo "Ovos"
161 df_final = pd.merge(df_postura, df_emerg, on="Postura", how="outer")
162 df_final = df_final.drop_duplicates()
163 # Descrição das colunas do data frame post_emerg_RAW:
164 # Datapostura = data da oviposição;
165 # Casal_AbP = identificação casal do tratamento e repetição proveniente da aba "Postura" do arquivo
Excel;
166 # Postura = identificação da postura;
167 # ndeovos_AbP = numero de ovos da postura proveniente da aba "Postura" do arquivo Excel;
168 # DataLeitura = data que foi avaliada a emergência das ninfas;
169 # ndeemerg = número de ninfas emergidas.
170 # 1. Salvar o DataFrame com todos os dados da aba postura e emergido para verificação
171 post_emerg_raw = os.path.join(diretorio, f"{nome_base}_post_emerg_RAW.csv")
172 df_final.to_csv(post_emerg_raw, sep=";", index=False, encoding="utf-8-sig")
173 # Visualizar as 5 primeiras linhas do DF e local de gravação do arquivo CSV
174 print("\nDados brutos das posturas e emergencia de ninfas:")
175 print(df_final.head())
176 print(f"\nArquivo com dados brutos das posturas e emergencia salvo como: {post_emerg_raw}")
177 # 2. Converter datas e valores numéricos corretamente
178 df_final["Datapostura"] = pd.to_datetime(df_final["Datapostura"], format="%d/%m/%Y",
errors="coerce")
179 df_final["ndeovos_AbP"] = pd.to_numeric(df_final["ndeovos_AbP"], errors="coerce")
180 df_final["ndeemerg"] = pd.to_numeric(df_final["ndeemerg"], errors="coerce")
181 # 3. Ordenar pelas datas de postura
182 df_final = df_final.sort_values(by="Datapostura")
183 # 4. Remover coluna "ndeovos(AbE)" se existir
184 if "ndeovos_AbE" in df_final.columns:
185     df_final = df_final.drop(columns=["ndeovos_AbE"])
186 # 5. Remover linhas com valores ausentes em ndeovos ou ndeemerg
187 df_final = df_final.dropna(subset=["ndeovos_AbP", "ndeemerg"])
188 # Agrupar os dados por casal
189 df_resumo = df_final[df_final["Postura"].str.startswith("ov", na=False)].copy()
190 # Converter datas para datetime para facilitar agregações
191 df_resumo["Datapostura"] = pd.to_datetime(df_resumo["Datapostura"], format="%d/%m/%Y",
errors="coerce")
192 # Agrupar por Casal(AbP)
193 df_agrupado = df_resumo.groupby("Casal_AbP").agg(
194     inicio_da_ov=("Datapostura", "min"),
195     final_de_ov=("Datapostura", "max"),
196     totovos=("ndeovos_AbP", lambda x: pd.to_numeric(x, errors="coerce").sum()),
197     totpost=("Postura", "count"),
198     totemerg=("ndeemerg", lambda x: pd.to_numeric(x, errors="coerce").sum())
199 ).reset_index()
200 # Agrupar os dados por casal
201 df_resumo = df_final[df_final["Postura"].str.startswith("ov", na=False)].copy()
202 # Converter datas para datetime para facilitar agregações
```

```

203 df_resumo["Datapostura"] = pd.to_datetime(df_resumo["Datapostura"], format="%d/%m/%Y",
errors="coerce")
204 # Agrupar por Casal(AbP)
205 df_agrupado = df_resumo.groupby("Casal_AbP").agg(
206     inicio_da_ov=("Datapostura", "min"),
207     final_de_ov=("Datapostura", "max"),
208     totovos=("ndeovos_AbP", lambda x: pd.to_numeric(x, errors="coerce").sum()),
209     totpost=("Postura", "count"),
210     totemerg=("ndeemerg", lambda x: pd.to_numeric(x, errors="coerce").sum())
211 ).reset_index()
212 # Agrupar os dados de total de ovip total de emergencia inicio da oviposicao e final de oviposicao por
casal
213 df_resumo = df_final[df_final["Postura"].str.startswith("ov", na=False)].copy()
214 # Converter datas para datetime para facilitar agregações
215 df_resumo["Datapostura"] = pd.to_datetime(df_resumo["Datapostura"], format="%d/%m/%Y",
errors="coerce")
216 # Agrupar por Casal_AbP
217 df_agrupado = df_resumo.groupby("Casal_AbP").agg(
218     inicio_da_ov=("Datapostura", "min"),
219     final_de_ov=("Datapostura", "max"),
220     totovos=("ndeovos_AbP", lambda x: pd.to_numeric(x, errors="coerce").sum()),
221     totpost=("Postura", "count"),
222     totemerg=("ndeemerg", lambda x: pd.to_numeric(x, errors="coerce").sum())
223 ).reset_index()
224 # Adicionar a coluna Sexo com "FEMEA"
225 df_agrupado['Sexo'] = 'FEMEA'
226 # Cria cópia apenas com Casal_AbP e Sexo = "MACHO"
227 df_macho = df_agrupado[['Casal_AbP']].copy()
228 df_macho['Sexo'] = 'MACHO'
229 # 3. Concatena os dois DataFrames
230 df_agrupado = pd.concat([df_agrupado, df_macho], ignore_index=True)
231 arquivo_resumo = os.path.join(diretorio, f"{nome_base}_post_emerg_RES.csv")
232 df_agrupado.to_csv(arquivo_resumo, sep=";", index=False, encoding="utf-8-sig")
233 # Descrição das colunas do data frame post_emerg_RES:
234 # Casal_AbP = identificação casal do tratamento e repetição proveniente da aba "Postura" do arquivo
Excel
235 # inicio_da_ov = data da primeira postura
236 # final_de_ov = data da última postura
237 # totovos = somatória de todos os ovos depositados pela fêmea
238 # totpost = somatória de todas as posturas depositadas pela fêmea
239 # totemerg = somatória de todas as ninfas emergidas das posturas depositadas pela fêmea
240 # Sexo = para verificação da consistência das informações do casal, esta coluna garante que uma
linha para cada sexo de cada casal seja criada, permite verificar se alguma fêmea não ovipositou, ou se a
morte de um macho não foi registrada.
241 # Visualizar as 5 primeiras linhas do DF e local de gravação do arquivo CSV
242 print("\nResumo por casal:")
243 print(df_agrupado.head())
244 print(f"\nArquivo salvo como: {arquivo_resumo}")

245 # *** Algoritmo 3 organiza os dados para o cálculo de longevidade, periodo de pré-oviposição etc.
246 # Ler a aba "casais" e "posturas" do Excel
247 df_excelcas = pd.read_excel(arquivo_excel, sheet_name="casais", header=None, dtype=str)
248 df_excelpost = pd.read_excel(arquivo_excel, sheet_name="posturas", header=None, dtype=str)
249 # Verificar e converter apenas os valores que são datas

```

```
250 df_excelcas = df_excelcas.applymap(lambda x: pd.to_datetime(x, errors='coerce').
strptime('%d/%m/%Y') if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
251 df_excelpost = df_excelpost.applymap(lambda x: pd.to_datetime(x, errors='coerce').
strptime('%d/%m/%Y') if pd.to_datetime(x, errors='coerce', format='%Y-%m-%d') is not pd.NaT else x)
252 # Processar aba "casais" esta aba informa quais adultos emergidos da fase ninfal foram utilizadas
para formação de casais
253 df6 = df_excelcas
254 df6 = df6.rename(columns={0: 'Original'})
255 # Inicialização
256 data_atual = None
257 casal_atual = None
258 resultados = []
259 i = 0
260 while i < len(df6):
261     item = df6.iloc[i, 0]
262     if '/' in item and len(item) == 10:
263         data_atual = item
264         i += 1
265         continue
266     if item.startswith("CAT"):
267         casal_atual = item
268         i += 1
269         continue
270     if item.startswith("T") and i+1 < len(df6) and df6.iloc[i+1, 0] in ["MACHO", "FEMEA"]:
271         trat = item
272         sexo = df6.iloc[i+1, 0]
273         resultados.append({
274             "Datacasal": data_atual,
275             "Casal_AbC": casal_atual,
276             "Trat_AbC": trat,
277             "Sexo_AbC": sexo
278         })
279         i += 2
280     else:
281         i += 1
282 # Criar DataFrame
283 df_casais = pd.DataFrame(resultados)
284 # Processar da aba "postura" a data da morte do inseto
285 df7 = df_excelpost
286 df7 = df7.rename(columns={0: 'Original'})
287 linhas = df7["Original"].tolist()
288 # Variáveis de controle
289 dados = []
290 data_atual = None
291 identificacao_atual = None
292 sexo_atual = None
293 # Percorre cada linha
294 for linha in linhas:
295     if '/' in linha: # Detecta data
296         data_atual = linha
297     elif linha.startswith('CAT') or linha.startswith('T1') or linha.startswith('CA'): # Identificação
298         identificacao_atual = linha
299         sexo_atual = None # Reseta o sexo
300     elif linha in ['MACHO', 'FEMEA']: # Sexo
```

```

301     sexo_atual = linha
302 elif linha == 'MORTO': # Ignora "MORTO"
303     continue
304 else:
305     # Outra informação irrelevante (como "ov..." ou números), ignora
306     continue
307 # Se já temos data, identificação do sexo, adiciona à lista
308 if data_atual and identificacao_atual and sexo_atual:
309     dados.append((data_atual, identificacao_atual, sexo_atual))
310     sexo_atual = None # Depois de adicionar, reseta sexo para aguardar um novo
311 # Cria um DataFrame com a data da morte do inseto
312 dfmort = pd.DataFrame(dados, columns=['Datamorte', 'Casal_AbP', 'Sexo_AbP'])
313 # Remove linhas onde Sexo está vazio (não é "MACHO" ou "FEMEA")
314 dfmort = dfmort[dfmort['Sexo_AbP'].notna()]
315 # Remove linhas duplicadas
316 dfmort = dfmort.drop_duplicates()
317 # Carregar os arquivos CSV gerados da fase ninfal para integrar os dados
318 df8 = pd.read_csv(os.path.join(diretorio, f"{nome_base}_fase_ninfal.csv"), sep=';')
319 # 1. Unir df_casais com df8 = _fase ninfal
320 df_merge1 = pd.merge(df_casais, df8, left_on="Trat_AbC", right_on="Trat_AbN", how="outer")
321 # 2. Unir o df anterior com dfmort (que reúne as datas de morte)
322 df_merge2 = pd.merge(df_merge1, dfmort, left_on=['Casal_AbC', 'Sexo_AbC'], right_on=['Casal_AbP', 'Sexo_AbP'], how='outer')
323 # Carregar o arquivo CSV post_emerg_res que congrega dos dados resumidos de posturas de cada
324 # femea
325 df9 = pd.read_csv(os.path.join(diretorio, f"{nome_base}_post_emerg_res.csv"), sep=';')
326 # 3. Unir df_casais com do arquivo post_emerg_res
327 df_merge3 = pd.merge(df_merge2, df9, left_on=["Casal_AbC", "Sexo_AbC"], right_on=["Casal_AbP", "Sexo"], how="outer")
328 df_final = df_merge3
329 # Nome do arquivo final
330 arquivo_final = os.path.join(diretorio, f"{nome_base}_fase_adult.csv")
331 df_final.to_csv(arquivo_final, sep=";", index=False, encoding="utf-8-sig")
332 # Limpa o df_final selecionando apenas as colunas desejadas
333 col_selec = ["Casal_AbC", "Datacasal", "Trat_AbC", "Sexo_AbC", "Trat_AbN", "Dataemerg",
334 "Datamorte", "Sexo_AbN",
335 "Peso", "Casal_AbP_x", "Sexo_AbP", "inicio_da_ov", "final_de_ov",
336 "totpost", "totovos", "totemerg", "Sexo"]
337 df_final = df_final[col_selec]
338 # Descrição das colunas do data frame final "fase_adult":
339 # AbN = dados proveniente da aba "Ninfa" do arquivo Excel
340 # AbA = dados proveniente da aba "Adulto"
341 # AbC = dados proveniente da aba "Casais"
342 # AbP = dados proveniente da aba "Postura"
343 # AbE = dados proveniente da aba "Emergidos"
344 # Datacasal = data da formação do casal
345 # Casal_AbP = identificação do casal
346 # Trat_AbC = inseto original usado para formar o casal
347 # Sexo_AbC = identificação do sexo da inseto
348 # Trat_AbN = identificação do inseto original
349 # Dataemerg (AbN)= data da emergência do adulto
350 # Datamorte (AbP)= data da morte do adulto
351 # Peso (AbA) = massa do adulto no primeiro dia em grama
352 # inicio_da_ov (AbP) - data da primeira postura

```

```
351 # final_de_ov (AbP) - data da ultima postura
352 # totpost (AbP) - total de posturas de cada femea
353 # totovos (AbP) - total de ovos de cada femea
354 # totemerg (AbE) - total de ninfas emergidas das posturas
355 # Sexo = para verificação da consistência das informações do casal, esta coluna garante uma linha
para cada sexo para cada casal seja criada, permite verificar se alguma fêmea não ovipositou, ou se a morte
de um macho não foi registrada.
356 # Nome do arquivo final
357 arquivo_final = os.path.join(diretorio, f"{nome_base}_fase_adult.csv")
358 df_final.to_csv(arquivo_final, sep=";", index=False, encoding="utf-8-sig")
359 # Visualizar as 5 primeiras linhas do DF e local de gravação do arquivo CSV
360 print(df_final.head())
361 print(f"\nArquivo salvo como: {arquivo_final}")
```