Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DO SOFTWARE

Reenxeñaría dun programa de pesado de colorantes

Estudante: Gabriel Alejandro López Schmidt

Dirección: Laura Milagros Castro Souto

A Coruña, setembro de 2023.

Dedicado a toda a xente que cando tiven momentos de dificultade tiñan momentos para polo menos envíar unha mensaxe de ánimo. Para os meus pais e irmás, que cando todo cambia, e non a mellor, sempre están aí, en especial Rocío, que día si, día tamén, estaba aí para escoitar a miña tristura feita palabra.

Agradecementos

Agradecer a tódolos profesores deste curso ponte o apoio e comprensión dada-las miñas circunstancias familiares e profesionais. E particularmente a Laura Castro por aceptar dirixir este proxecto, amais do que aprendín e desfrutei na materia de Arquitectura do Software.

Resumo

A firma italiana Termoelettronica ten un conxunto de software para o control de maquinaria de tinturaría de prenda confeccionada. Cada software é específico para una tarefa, e existe un para o pesado de colorante, normalmente sólido (en pó), mediante control de báscula. Dito programa, recibe unha orde de pesado e baseándose nunha base de datos de colorantes e mediante escaneado do código de barras e a medida dunha báscula compatible, permite a pesada dunha fórmula de cor. Unha vez aprobada a pesada, o sistema da luz verde e pode xerar un recibo de confirmación. Cada vez que a pesada é validada, almacénase esa información (colorantes implicados, cantidades, pesador, tempo) para poder continuar co proceso produtivo. Este programa actualmente non funciona nin está integrado na liña de produción de Lavamar S.A., o que na práctica significa que o sistema de pesada é manual sen ningún tipo de control nin auditoría, alén da propia capacidade do pesador para facer ben o traballo. Así pois, o obxectivo deste Traballo Fin de Grao é desenvolver un substituto dese software para integralo nun sistema de produción heteroxéneo, con participación de sistemas da empresa SETEX, a propia Termoelettronica, TregiSistemi, e outros.

Abstract

The Italian firm Termoelettronica provides a set of software for controlling machinery in the dyeing process of manufactured garments. Each software program is specific to a particular task, and there is one for weighing dye, typically in solid form (powder), using a scale.

This program receives a weighing order and, based on a database of dyes, scans the barcode and measures the weight using a compatible scale, enabling the weighing of a color formula. Once the weighing is approved, the system gives a green light and can generate a confirmation receipt. Every time the weighing is validated, the information (dyes involved, quantities, weigher, time) is stored to continue with the production process.

Currently, this program does not work, nor is it integrated into the production line of Lavamar S.A.. In practical terms, this means that the weighing system is manual without any control or auditing beyond the weigher's ability to perform the task correctly. Therefore, the objective of this Degree Project is to develop a reengineered substitute for this software to integrate it into a heterogeneous production system involving systems from companies like SETEX, Termoelettronica, TregiSistemi, and others.

Palabras chave:

- Tinte
- Tintura
- Colorante
- Pesaxe
- Báscula
- Balanza
- Receita
- Tricomía
- Paradox
- Planificación

Keywords:

- Dye
- Dyeing
- Weighing
- Scale
- Balance
- Recipe
- Trichomy
- Paradox
- Planning

Índice Xeral

Índice de Figuras

Índice de Táboas

Capítulo 1

Introdución

O presente Traballo de Fin de Grao aborda a reenxeñaría dun programa de pesado de colorantes na industria téxtil. A xustificación deste proxecto radica na necesidade de soluciona-las limitacións e problemas existentes co software que debería ser usado a tal fin, pero que por distintos motivos (adaptabilidade ós sistemas actuais, obsolescencia, falta de integración,...) non se usa, aínda que por unha cuestión de trazabilidade e cumprimento duns mínimos estándares de calidade, debería ser usado para unha correcta certificación das pesadas. Así mesmo, queremos abordar tamén con este traballo unha procura de mellores prácticas e eficiencia no proceso de tinturaría de prendas confeccionadas.

1.1 Contexto de negocio: o proceso de tintura industrial

Lavamar S.A é unha empresa da Coruña que se dedica ó negocio dos acabados téxtiles sobre prenda confeccionada. A súa actividade principal son os procesos de lavado e tintura, mais nos últimos tempos diversificou a súa actividade co emprego de novas tecnoloxías como son o láser, ozono, nebulización e estampación dixital. A día de hoxe, conta con máis de 30 anos de experiencia e é un referente no sector, xa que traballa con empresas punteiras no campo téxtil.

Lavamar conta cunhas instalacións dotadas de distintas tecnoloxías e sistemas de automatización que deben ser integrados para conseguir unha xestión integral non só do proceso produtivo, senón tamén de calquera aspecto que afecte á trazabilidade do traballo realizado.

O proceso principal en Lavamar é a tintura de prendas. Para iso fan falta unha serie de elementos que deben estar perfectamente controlados:

- *Maquinaria*: Úsanse principalmente máquinas de tintura de distintas marcas (Tupesa, JMC, ...) aínda que sempre abertos á innovación, como será o uso de máquinas de Jeanología.

- *Colorantes*: As distintas cores lógranse co uso de distintos colorantes nunhas proporcións concretas, que se obteñen a partir do traballo do laboratorio, onde se proban as novas producións que piden os clientes.
- *Produtos químicos*: Para que os colorantes actúen dun xeito adecuado, é preciso controla-la súa absorción polas fibras mediante o uso de distintos produtos químicos, dende drogas estandarizadas (ácido acético, hipoclorito sódico, ...) até solucións privativas de distintos fornecedores.
- *Coñecemento adquirido*: Como o obxectivo final é o acabado en prenda confeccionada, hai que coñecer moi ben como vai reaccionar non só o tecido da roupa, senón a roupa en si, xa que non é o mesmo a tintura dunha camisa de manga longa que se tivese manga curta. Iso fai que haxa unha serie de condicionantes que hai que aplicar para obter un resultado satisfactorio.

Todos estes elementos únense para o proceso industrial denominado *tintura sobre prenda confeccionada*, que aínda que pode ter moitísimas variantes, por regra xeral segue un fluxo definido que describimos a continuación:

1. *Recepción, identificación e preparación da roupa*: A roupa que se recibe debe ser correctamente rexistrada en base ás súas características:
   - Cliente.
   - Tempada: Coa tempada refírese un período específico no que se presentan e comercializan as coleccións de roupa.
   - Artigo: Identifica o modelo definido por un tipo de prenda e pertencente a unha categoría xenérica: home, muller, bebé...
   - Calidade: Identifica o tecido do que está feito o artigo e tamén como está confeccionado. É unha das cousas que máis inflúen no proceso de tintura.
   - Cor: O que dá sentido a esta industria é acada-la cor solicitada polo cliente sobre o artigo e calidades antes indicadas.

   A preparación da roupa tamén é algo importante: Non é o mesmo tinxir camisas de manga longa que petos de bebé, ou pantalóns vaqueiros que uns de pana. Todo é algodón, pero nas máquinas reaccionan distinto. Así, por exemplo, ás camisas de manga longa hai que suxeitarlle-las mangas.
2. *Obter Receita de tintura*: Se a roupa é a primeira vez que entra no sistema produtivo, a nosa sección de Laboratorio ten que intentar acadar a cor solicitada até conseguila aprobación do cliente. Isto faise de xeito iterativo con probas sobre retais a partir

normalmente dunha mostra enviada polo mesmo cliente. Nese momento teremos unha fórmula enviada a produción.

3. *Fraccionar roupa e planificar*: As prendas que entran deben ser particionadas en cantidades que poidan ser tintadas polas máquinas de produción. Un peso medio poderían ser 150 kg de roupa por máquina. Esa partición de roupa debe planificarse nunha máquina, nun día e nunha orde concreta. E esa roupa que está plenamente identificada, ten unha Receita adscrita: a Receita de produción.

4. *Proceso de tintura*: Cando a roupa é tintada nunha máquina, segue unha serie de instrucións explicitadas nunha Receita personalizada. Ditas instrucións van dende indica-lo número de voltas por minuto, os litros de auga cos que hai que enche-la máquina, cando baleirala e tamén cando, como e con que cantidade hai que bota-los produtos químicos e os colorantes. A roupa sufre unha serie de accións físico-químicas, que provocan que o colorante sexa absorbido pola fibra, de forma homoxénea, e que despois non se perda. Ademais tamén é importante o acabado ó tacto final, pola acción dos suavizantes. A acción física vén definida pola acción mecánica da máquina coas voltas por minuto e as inversións; a auga vai variando a súa temperatura dependendo o tipo de proceso e en que fase da tintura esteamos. Tamén inflúe moito a forma do bombo da máquina: se é aberto ou compartimentado. A acción química vén definida polo uso de produtos químicos e colorantes. Algúns dos primeiros preparan a roupa para recibi-lo colorante e fixalo ás fibras. Outros teñen outros cometidos. A interacción de todo o conxunto provoca, se todo é correcto, o resultado desexado.

5. *Turbinado, secado e control de calidade*: Despois da tintura, a roupa pasa por unha turbina para deixa-la coa menor auga posible antes de entrar nunhas grandes secadoras. Coa roupa enxoita, faise un exhaustivo control de calidade para verificar que non haxa danos na roupa, e que a cor é a solicitada polo cliente.

Figura 1.1: Liña de tinte de Lavamar

1.1.1 Problemas actuais. Situación de partida

Nesta sección, abordaremos os desafíos e problemáticas actuais que nos levan a propoñer este proxecto. É fundamental comprender a situación de partida para poder abordar de maneira efectiva e atopar solucións a estes problemas. A través dunha análise detallada, avaliaremos as posibles estratexias para superar estas dificultades. Esta sección proporciona unha base sólida para a discusión e resolución dos problemas enumerados:

- Falta de trazabilidade na operación de pesaxe. Só existe a sinatura do pesador. Isto implica que non sabemos realmente o que pesa o pesador. Só sabemos o que ten que pesar. A única verificación posible de que pesou ben e cando a roupa xa está en etapas posteriores do proceso de tintura.
- Ante cores novas úsase un sistema de verificación de dobre comprobación: dous pesadores a un tempo. Isto implica que unha persoa está controlando a outra para confirmar que está pesando correctamente. Volvemos á falta de trazabilidade: quen di que o supervisor non está errado?
- Ante un problema de pesaxe, ben porque non se pesaron as cantidades indicadas, ben porque se equivocaron ó escolle-lo colorante, o resultado na gran maioría das ocasións será unha reoperación das prendas co que iso significa en custes de enerxía, tempo, persoal, auga e produtos químicos. Ademais, a roupa pode sufrir danos. Un problema de pesaxe é case sempre un problema humano, xa que o fallo nas básculas é improbable debido a que teñen que confirmar o seu funcionamento con cada uso.
- Ter un software/hardware adquirido previamente sen usar.

Partimos dun sistema de axuda á pesaxe guiada de colorantes, que segundo as súas especificacións podería soluciona-los problemas indicados previamente. Este sistema guiaría ó pesador ó longo de todo o proceso de pesaxe de colorantes, indicando paso a paso as actuacións do pesador, o colorante a empregar mediante código, o peso obxectivo e o peso actual, permitíndolle valida-la pesaxe cando estes fosen iguais. Ademais, gardaría un rexistro das pesaxes efectuadas. Dito sistema era sabido que existía pero nunca se usara. Estaba nun almacén con evidentes signos do paso do tempo.

Conseguiuse recupera-lo programa dunha tarxeta SD que estaba conectada á placa do sistema, xa que o *backup* que se fixera no seu tempo estaba mal nomeado e referíase a outro programa do sistema de Termoelettronica.

O equipo orixinal estaba nun estado inoperativo. Ademais de ser un sistema empotrado, non lle funcionaba a fonte de alimentación. Empregaremos outro ordenador que había na empresa para a execución do programa: TermoPowderM.

(a) Armario orixinal para o sistema de pesaxe

(b) Detalle placa do sistema orixinal

(c) Detalle do interior do armario de pesaxe

1.2 Obxectivos

Co proceso de reenxeñaría do programa existente e a súa potencial integración no sistema de produción, buscase non só optimizar os recursos existentes, senón tamén mellorar a precisión e reproducibilidade do pesado de colorantes, incrementar a eficiencia do proceso e proporcionar un maior control e seguimento da pesada. Con isto se evitarían as reoperacións de prendas e as perdas a nivel de custes de persoal, tempo, auga, enerxía e uso de químicos.

Ademais, conseguiriamos a amortización dun sistema xa adquirido polo propio persoal da empresa pero sen uso real até a data. Un sistema novo, cun sistema de seu, máis a adaptación ás básculas existentes, custaría ó redor de 6.000 euros (ver orzamento do 2020, incluído no apéndice A) e mesmo non aseguraría o seu uso nun entorno heteroxéneo con maquinas de distintos fabricantes. Grazas a este proxecto recuperase para o seu uso un PC táctil que estaba almacenado sen destino definido, así como un software que nunca fora utilizado.

Este proxecto pretende ofrecer unha solución viable e eficaz para o seguimento e rexistro do pesado de colorantes e a súa integración no sistema produtivo de Lavamar, contribuíndo así á mellora da calidade e competitividade no sector téxtil.

1.2.1 Obxectivos concretos

Búscase a reenxeñaría dun programa de control de pesado de colorantes e a súa integración co sistema actual de tintura de prendas. En concreto, o resultado do proxecto debe achegar:

- Reproducibilidade. Evitar reoperacións. A pesaxe correcta do colorante asegura nunha porcentaxe moi alta que a cor que quere o cliente vaise poder reproducir. Sempre que non é así, hai que reopera-la roupa para volve-la aproximar á cor obxectivo.
- Trazabilidade das pesadas para eleva-los estándares de calidade.
- Eleva-lo nivel de automatización da fase de pesaxe e mellora-la súa integración no fluxo de traballo.
- Facilita-la supervisión da pesada.
- Elimina-la necesidade de dobre pesador para novos pedidos.
- Facilita-la formación de novos pesadores.
- Aforro económico. A reenxeñería evita a adquisición dun sistema novo que non necesariamente sería fácil de integrar.

A nivel técnico o piar fundamental vai ser a reutilización por integración, que debe proporcionar:

- Combinación de elementos propios (*legacy*) e adquiridos a terceiros (COTS).
- Execución en diferentes máquinas físicas, plataformas diferentes e en distintas ubicacións.
- Control sobre os diferentes sistemas involucrados.
- Minimización de dependencias sen rachar a integración. Adaptación do software propio.
- Uso de XML como formato de intercomunicación pola alta flexibilidade para nutri-las fontes de datos.
- Maximización da dispoñibilidade.
- Integración de datos para que cada elemento a integrar funcione como foi pensado.
- Comunicación remota coas fontes de datos.

1.3 Estrutura da memoria

Nese último punto menciónanse como están estruturados o resto de capítulos da memoria, para anticipar brevemente o contido de cada un deles.

- *Xestión do desenvolvemento software* (capítulo 2), no que se trata a organización e dirección do proxecto de desenvolvemento, incluíndo metodoloxías e ferramentas de xestión.
- *Análise do sistema* (capítulo 3), no que analizaremos polo miúdo o sistema a desenvolver e toda a información dispoñible, tanto a nivel de arquitectura como a nivel de compoñentes.
- *Análise de requisitos* (capítulo 4), onde analizaremo-las necesidades funcionais e non funcionais que debe cumpri-lo sistema para asegurar o funcionamento desexado.
- *Deseño* (capítulo 5), onde daremos forma a unha arquitectura obxectivo detallada e a unhas directrices de implementación.
- *Implementación e probas* (capítulo 6), onde se codificará todo o planeado nos pasos anteriores, e faranse probas para verifica-lo seu funcionamento.

- *Integración e validación* (capítulo 7), capítulo no que se describe como se integran os distintos elementos e as probas para valida-lo seu funcionamento.
- *Despregamento, posta en marcha e mantemento* (capítulo 8), no que describiremos a inserción do sistema en produción, dando pautas para o seu mantemento.
- *Conclusión* (capítulo 9), capítulo no que describiremo-lo resultado final do proxecto, facendo mención ás leccións aprendidas e posibles liñas de actuación futuras.

Capítulo 2

Xestión do desenvolvemento software

Neste capítulo, abordaremos unha parte fundamental para asegurar que o software se desenvolva de maneira eficiente, cumpra cos requisitos do cliente e se entregue dentro do prazo e orzamento establecidos. A xestión do desenvolvemento de software é o conxunto de prácticas, procesos e técnicas utilizadas para planificar, coordinar e supervisar o desenvolvemento dun proxecto de software desde a súa concepción ata a súa entrega final. Inclúe a definición de obxectivos, a asignación de recursos, a planificación de tarefas, a supervisión do progreso e a toma de decisións para asegurar a calidade e o éxito do proxecto.

2.1 Metodoloxía de traballo

Debido á escasa documentación da que se dispón no momento de arrancar o proxecto, aplicaranse prácticas inspiradas nas metodoloxías áxiles, que chegado o caso poden derivar no uso de prototipos e iteracións rápidas. A flexibilidade e adaptablidade promovidas polas tecnoloxías áxiles combinan ben con este proxecto, onde ante unha contorna moi heteroxénea, precísase a capacidade de facer axustes e cambios no enfoque ou nos plans conforme sexa necesario.

O uso de prototipos pode axudar a validar ideas e recoller información ante un escenario incerto como o que afronta este proxecto.

As iteracións rápidas permitirán que facendo pequenos incrementos de funcionalidade ou melloras, se obteña unha retroalimentación constante e permitirá visualizar mellor a evolución do desenvolvemento do proxecto.

Os coñecementos previos sobre o fluxo de traballo e as necesidades operativas, permitirán que o sofware se desenvolva de acordo as expectativas.

O proceso de análise pode requirir varias revisións e refinamentos a medida que se gaña

unha comprensión máis profunda do sistema.

Seguindo as pautas para a reconstrución/reenxeñería de software, e para manter os parámetros de calidade e cumprir cos distintos requirimentos realizarase a seguinte secuencia de actividades:

- *Análise detallada do sistema*: A análise implica unha comprensión en profundidade que pode incluír a interpretación dos elementos do sistema de maneira iterativa. Isto significa que o proceso de análise pode requirir varias revisións e refinamentos á medida que se gaña unha comprensión máis profunda do sistema.
  - Proceso interpretativo, interactivo e iterativo, para comprender a fondo o sistema
  - Tomar decisións arquitecturais.
  - Avaliar mantemento e reconstrución do sistema, ou un novo desenvolvemento.
  - Identificación de elementos reutilizables
- *Extracción da información*: Colleita de toda a información relevante do sistema existente.
- *Identificación de fontes de datos*: Localizamos e identificamos as fontes de datos que alimentan o sistema.
- *Fusión de vistas*: Integramos e unificamos diferentes perspectivas para obter unha visión completa e coherente do sistema.
- *Reconstrución*: Reconstruímos un sistema que debería estar operativo.

2.2 Fases principais do traballo

O proceso de desenvolvemento de software require unha secuencia de pasos estruturados e definidos para asegurar a súa eficacia e éxito. Cada unha destas etapas xoga un papel crucial no resultado final:

1. Avaliación inicial: revisar calquera información relevante para determinar o estado do software e os requisitos dos usuarios. Isto proporciona unha base fundamental para que proxecto conclúa con éxito.
   - Análise do sistema:
     - Extracción da información, onde recollemos datos relevantes para o proxecto, como información existente, documentación e calquera outra fonte de datos que axude a comprender o contexto e as necesidades do sistema a desenvolver.

  - Identificación das fontes de datos. É importante identificar os consumidores e xeradores de información, de onde veñen e a onde van os datos necesarios para o proxecto: bases de datos existentes, sistemas externos, ficheiros...
  - Determinación da situación obxectivo, xa que é importante establecer con claridade as metas e obxectivos específicos que o software deberá alcanzar.

2. Análise de requisitos: Esta etapa implica a identificación e documentación detallada dos requisitos que debe verifica-lo noso sistema, nomeadamente:

   - Requirimentos funcionais, onde especificaremos o que o sistema debe facer.
   - Requirimentos non funcionais, onde o importante é o como se comporta o sistema a nivel de rendemento, fiabilidade e seguridade.

3. Deseño: Desenvolvemento dun plan de deseño para describir a estrutura e a funcionalidade que terá o software. Achegaránse diagramas e outro tipo de indicacións para que sexan empregados como guía na fase de implementación.

4. Implementación: Escritura de novo código e integración de código existente no novo deseño. Inclúe as probas de funcionamento e validación das distintas unidades de software.

5. Integración: Realízase a integración dos distintos elementos para validar o seu funcionamento coma un todo e probas que validen os requirimentos no sistema desenvolvido.

6. Despregamento: Instalación do software e hardware necesario para poder face-la posta en funcionamento en Lavamar.

7. Mantemento: Soporte para un funcionamento continuado axeitado, mediante o rexistro e solución de incidencias.

Para a realización deste proxecto escóllese un ciclo de vida de desenvolvemento en V[1], que foi escollido debido a súa adecuación ós requisitos e natureza do mesmo.

2.3 Estimación Temporal e Planificación

A fin de facilitar e monitorizar o progreso no desenvolvemento do proxecto, realízase a estimación temporal e a planificación do mesmo. Estas etapas axudan a determinar o tempo e os recursos necesarios para completar con éxito cada fase do proxecto. Ademais, achegan datos obxectivos para a toman de decisións a través da análise das tarefas, a complexidade do software, e as posibles incertezas.

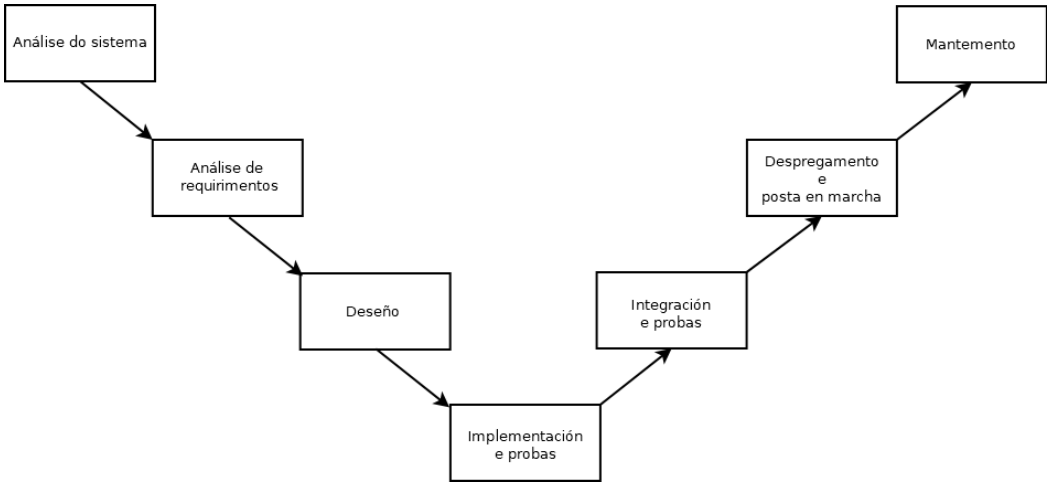


Figura 2.1: Ciclo de vida. Modelo en V

2.3.1 Estimación inicial

A estimación inicial implica unha avaliación preliminar dos recursos, tempos e esforzos para levar a cabo o proxecto con éxito. Mais neste caso non se realizará unha estimación inicial detallada, debido á natureza complexa e incerta do proxecto e debido ós compromisos persoais e profesionais, tamén moi flutuantes. Esta decisión é coherente co enfoque áxil adoptado que permitirá adaptarse as necesidades e evolución do proxecto.

En ausencia dunha estimación inicial, non se realiza tampouco unha planificación estrita, aínda que si se fixa como límite o marco temporal agardado para o desenvolvemento dun TFG no plan de estudos da titulación do Grao en Enxeñaría Informática [2], cun esforzo asociado de entre 12 créditos, estimado entre 25 e 30 horas de traballo do alumno por crédito, que nos ofrecen unha marxe de entre 300 e 360 horas de dedicación.

2.3.2 Temporalidade final

A presentación dos resultados da temporalidade final é fundamental para avaliar a eficacia da xestión do tempo e recursos ao longo do proceso. A través da análise dos datos reais rexistrados durante o proxecto, é posible gañar información valiosa sobre a duración de cada etapa, xa que a comprensión da temporalidade final é esencial para mellorar a precisión das estimacións en proxectos futuros e para axustar a planificación en consecuencia. Amosámolo cronograma deste proxecto en forma de diagrama de Gantt, onde de forma gráfica se resume o seguimento e tempos de execución deste proxecto. A través desta representación temporal, buscase destacar a secuencia de tarefas e proporcionar unha visión completa do proceso que levou á conclusión exitosa deste proxecto.

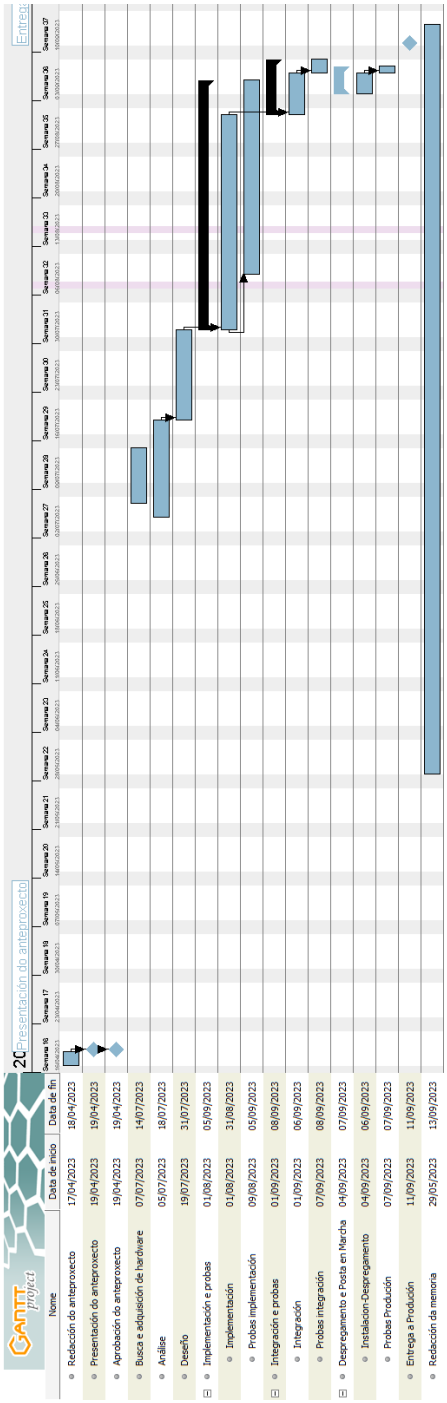


Figura 2.2: Cronograma do proxecto

2.3.3 Resumo do cronograma

Destacámo-los datos derivados do cronograma facendo un resumo dos mesmos:

Data de inicio do proxecto: 17/04/2023
Data de fin do proxecto: 11/09/2023

Tarefa	Horas-Persoa	Porcentaxe
Anteproxecto	9	3%
Busca e adquisición de hw	27	8%
Análise	45	14%
Deseño	41	13%
Implementación e probas	108	34%
Integración e probas	27	8%
Despregamento e posta en marcha	18	6%
Redación da memoria	45	14%
Total	**320**	**100**

Táboa 2.1: Temporalidade do proxecto

2.3.4 Estimación de custes

A partir dos datos extraídos da planificación do proxecto, vese que o total de esforzo necesario para completa-lo proxecto foi de 320 horas-persoa, que se corresponde co esforzo agardado dada a carga académica do TFG, de 12 créditos.

Para poder asignar un valor económico ó desenvolvemento deste proxecto temos que fixar un valor o custe de hora-persoa. Se o facemos, por exemplo, cun valor *custe hora-persoa*= 12€, podemos calcular o custe do proxecto, que queda fixado en 3840€, que vemos que un custe moi competitivo con respecto do amosado no anexo A.

Capítulo 3

Análise do sistema

O propósito principal deste capítulo é proporcionar unha visión completa e detallada do sistema que estamos a abordar. Isto implica unha análise exhaustiva de tódolos aspectos relacionados co sistema, dende a súa arquitectura xeral ata os seus compoñentes individuais e as súas interaccións. A través desta análise, buscarase asegurar que o noso proxecto se adapte eficazmente ás necesidades e obxectivos definidos na etapas iniciais do proxecto.

3.1 Reenxeñaría da arquitectura

Dado que este capítulo segue os pasos habituais para unha reenxeñaría (ilustrados na figura 3.1, imos facer unha breve introdución de cada un:

- *Avaliación inicial*: Inicialmente, realizamos unha avaliación completa do sistema existente. Iso inclúe a revisión de toda a información relevante dispoñible para determinar o estado actual do software e identificar os requisitos de usuario.
- *Extracción da información*: Procedemos á extracción de informacións críticas do sistema existente, como a recollida de datos e detalles relevantes para o proceso de reenxeñaría.
- *Identificación das fontes de datos. Consumidores e produtores*: Este paso é crucial para entender as diferentes fontes de datos que alimentan o sistema. Iso inclúe a identificación tanto dos consumidores como dos produtores de datos, fornecendo unha visión completa do ecosistema de información.
- *Situación actual. Arquitectura*: Nesta etapa, analizamos e documentamos a arquitectura actual do sistema. Iso inclúe a comprensión das interaccións entre os diferentes compoñentes, a identificación de posibles áreas de mellora e a avaliación da estrutura xeral.

- *Fusión de vistas. Hipótese.*: Coa análise de toda a información, podemos ter en conta múltiples puntos de vista, múltiples visións do problema e formular unha hipótese para avanzar na solución.
- *Arquitectura tentativa. Reconstrución*: A partir de tódolos pasos anteriores deseñamos unha arquitectura tentativa que nos leve, sabendo que a reenxeñaría é interpretativa, interactiva e iterativa, cara unha reconstrución do sistema. Este punto verémolo no capítulo de deseño (capítulo 5, páxina 37).

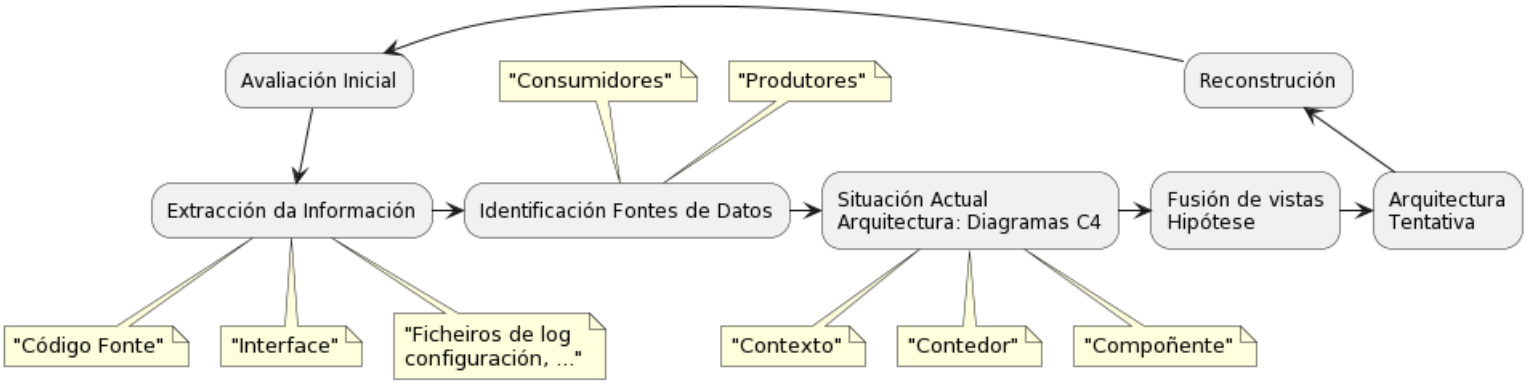


Figura 3.1: Secuencia dunha reenxeñaría da arquitectura

3.2 Avaliación inicial: revisión de documentación

Como primeiro paso, procedemos á revisión de calquera información relevante para determina-lo estado do software e os requirimentos de usuario.

Na revisión de documentación existente atopamos un manual de usuario, traducido ó castelán con fotos da aplicación en italiano. Trátase dun manual de uso da aplicación. O manual amosa diferentes chamadas ó programa executable e distintas opcións de configuración por liña de comandos. O programa permite a configuración de varias balanzas, normalmente unha centesimal e outra decimal.

A sección máis interesante é a *páxina de estados*, onde fala da interface que aparece ó arrinca-lo programa (ver figura 3.2). Segundo este, visualízanse en tempo real os seguintes datos:

Datos do estado

- *Produto e descrición*: Código e nome do produto que está en fase de ser pesado.

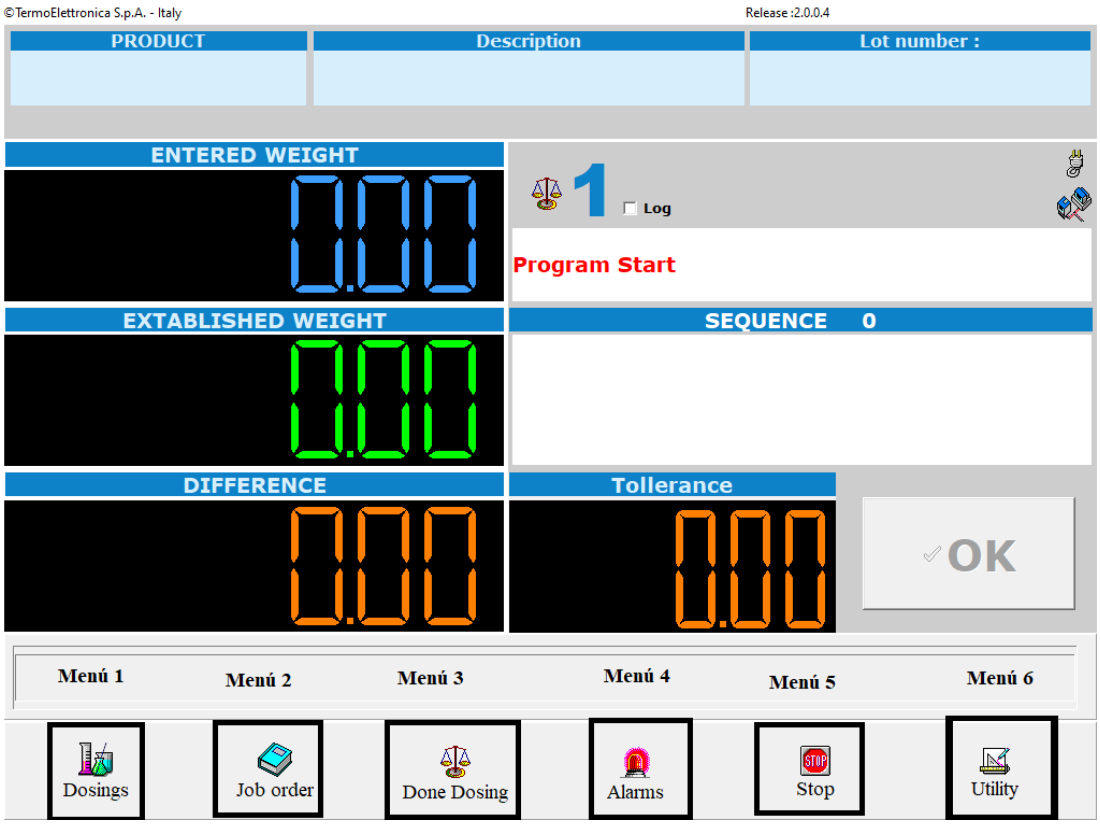


Figura 3.2: Interface do programa de pesaxe

- *Nº de pedido*
- *Número de lote*: Debería poñer pedido. Aparece o nº de pedido[1] que está en execución, cantos colorantes ten esa tricomia e a orde a que pertence o colorante que está a ser pesado.
- *Báscula*: Indica a báscula que está activa.
- *Estado*: Amosa estados do equipo: conectado á rede eléctrica, á rede LAN,...

Visualización de pesada en curso Aparece a cifra a pesar, a pesada en curso, a diferenza absoluta entrambas e tamén a diferenza en tanto por mil. Ademais de xeito visual, aparecen dúas liñas horizontais, unha representa analoxicamente a cantidade a pesar e a outra o valor do produto que se está dosificando. Vese tamén a tolerancia permitida.

Operacións suxeridas Trátase dunha área textual onde se indica ó pesador os posibles pasos a seguir:

- Sistema listo para efectua-lo pedido

[1] `IdMaquinada` na nomenclatura da empresa

- Lectura de código de barras
- Solicitar colocación dunha bandexa
- Agregar colorante á balanza
- Sacar colorante da balanza
- Peso dentro da tolerancia

Menú principal Acceso ó resto de opcións do programa, nomeadamente:

- *Xestión de dosificación directa* (figura 3.3). O manual indica soporte para:
  - Posibilidade de crear tricomías. sen planificación previa.
  - Posibilidade de modificar tricomías, previamente planificadas.
  - Borrado total ou parcial de tricomías.

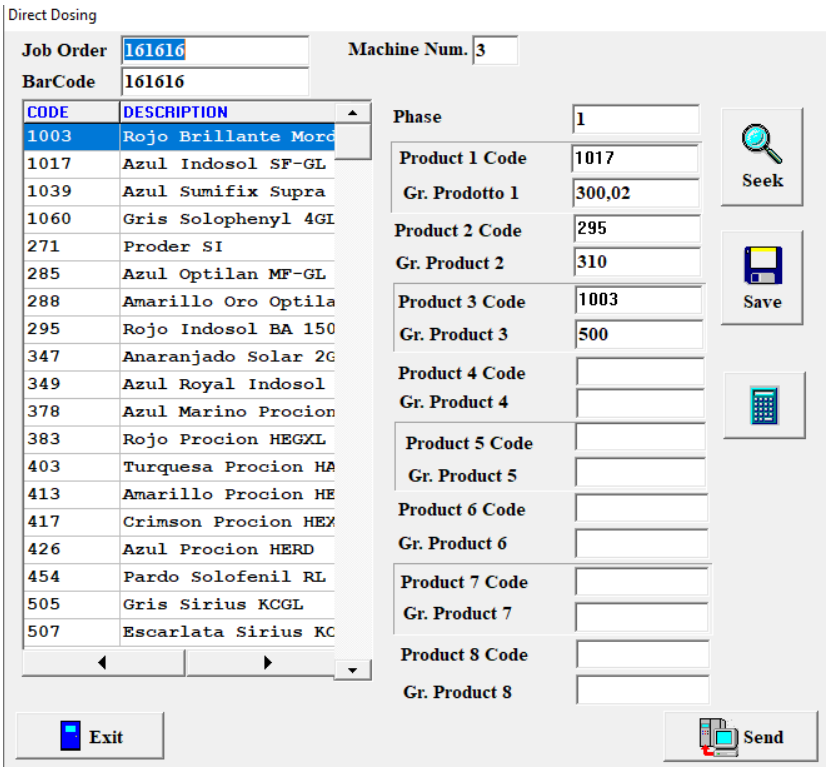


Figura 3.3: Menú 1. Xestión de tricomías e cantidades

O que podemos ver na imaxe son os seguintes datos:

- Código e descrición, onde figuran os colorantes rexistrados no almacén e o seu código.
- Inserción de datos para a composición dun pedido. Mediante os códigos de colorante e as cantidades a pesar queda definida a tricomía.

  - Nº de pedido e nº de máquina.
  - Código e cantidade de cada colorante.
  - Botón de busca, gardado e envío de pedido ó controlador para a pesaxe.
- *Xestión de pedidos.* Vista dos pedidos na lista de traballo. Os pedidos completados aparecen en negro. Vemos na imaxe 3.4 que aparece información tabulada do pedido, nº de máquina, data de inicio e data de fin. Vemos que non é un formulario moi ben deseñado pois non permite o reescalado para ver en conxunto tódalas columnas dun xeito cómodo, habendo que desprazarse cos botóns de dirección. Ó seleccionar unha fila aparece información sobre a tricomía a pesar, nomeadamente:
  - Fase: Indica o punto de introdución do produto. Os colorantes da mesma fase deben ir xuntos. Isto pasará sempre porque tódolos colorantes dunha tricomía teñen a mesma fase, forzado polo programa de formulación.
  - Código: Código do colorante.
  - Descrición: Nome oficial do colorante[2].
  - Cantidade en gramos.
  - Botóns e campos de xestión de pedidos: Busca, Cancelación, Ordenación, Envío a dosificación.
- *Histórico de pesaxes*: Visualización das pesaxes efectuadas (figura 3.5), coa seguinte información:
  - Pedido, código, descrición, data de execución da dosificación.
  - Peso medido: cantidade real pesada. Lembramos que non ten por que coincidilo peso medido co solicitado ó permitir o sistema un limiar de tolerancia na pesaxe.
  - Peso solicitado.
  - Botóns de borrado de pesadas efectuadas.

Xestión de alarmas O programa considera alarmas a calquera evento significativo durante a execución do programa: inicio e saída do programa, lote interrompido, tempo de balanza expirado,... De algúns destes eventos só rexistra cando se inician, noutros como no caso do tempo de balanza expirado rexístrase a hora cando se resolve a alarma (ver figura 3.6).

- Data de inicio: data e hora na cal se produciu a alarma.
- Nº de alarma.

[2] En Lavamar o nome coincide co indicado en The List by Inditex[3] ou no ZDHC[4]

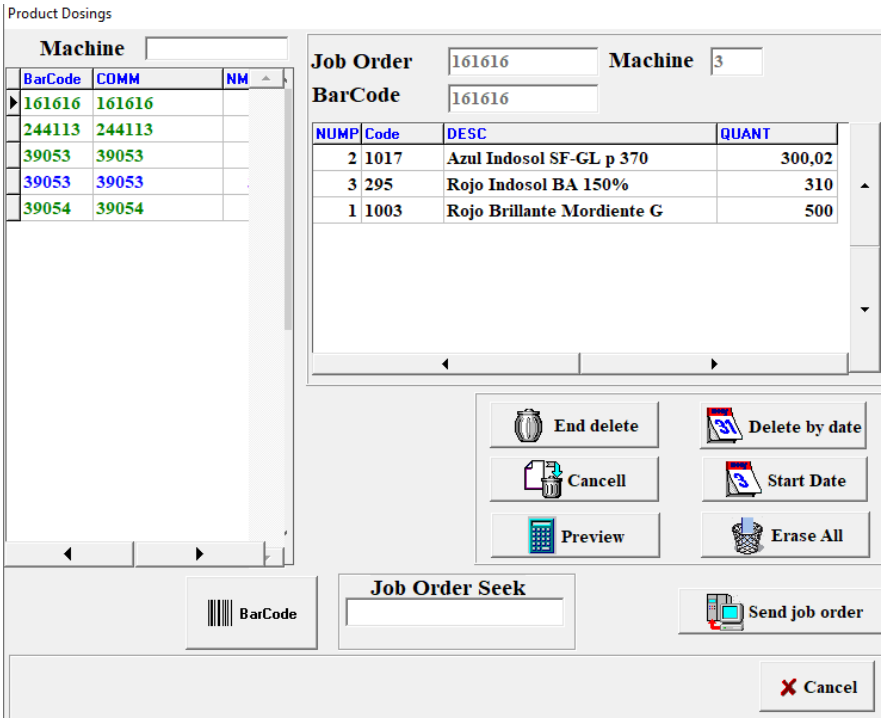


Figura 3.4: Menú 2. Lista de tricomías rexistradas

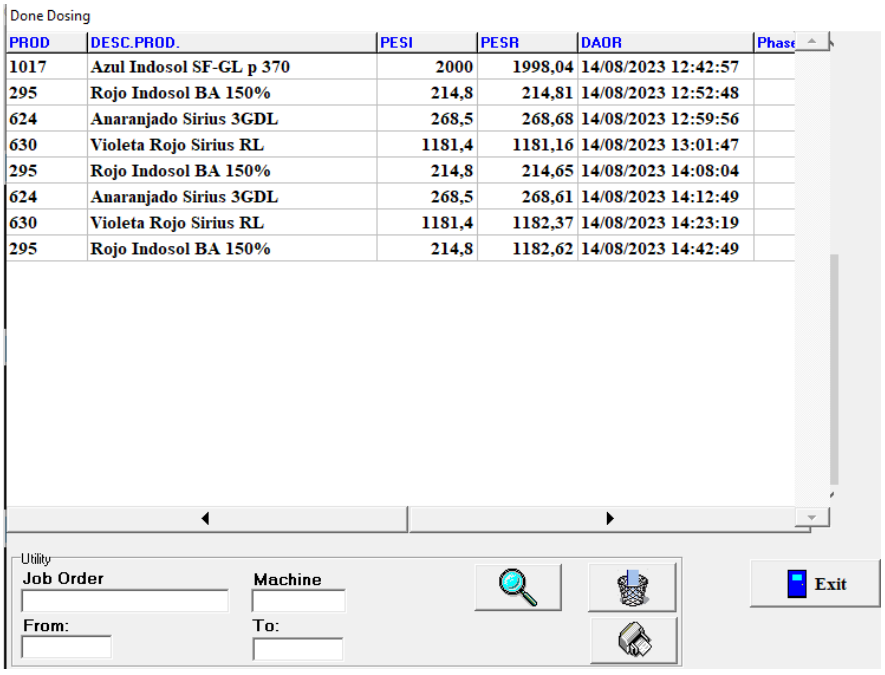


Figura 3.5: Menú 3. Pesaxes efectuadas

- Data final: data e hora na que se desactivou a alarma.
- Descrición da alarma.
- Botóns de borrado de alarmas.

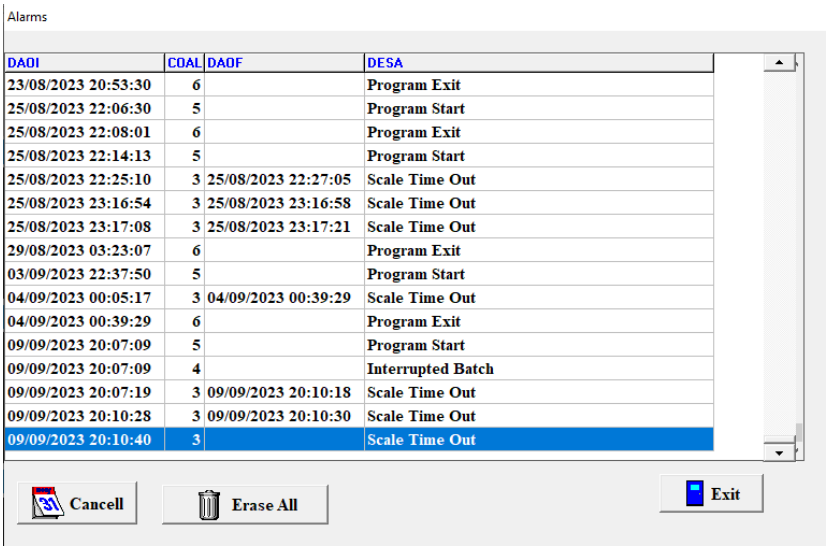


Figura 3.6: Menú 4. Lista de alarmas

Parada de elaboracións Dá varias opcións (figura 3.7):

- *Terminar*: Rematar elaboración en curso. Os produtos non dosificados, ignóranse.
- *Anular*: Anula-la dosificación do produto en curso. A elaboración restablécese co seguinte produto do pedido.
- *Espera*: Permite coloca-lo sistema en espera, para poder efectuar controis eventuais.

Utilidades Agrupa a configuración dunha serie de elementos heteroxéneos (figura 3.8), como por exemplo:

- *Parámetros fixos* (figura 3.9), entre os que destacan:
  - Nº DPA: Nº de identificación da estación de pesado
  - Kg de capacidade da balanza pequena e grande
  - Hectogramos límite que se poden pesar na balanza pequena.
  - Nº de balanzas conectadas
  - Tolerancia en ‰
  - Tipo de balanza: Mettler, Sartorious, And, Gib

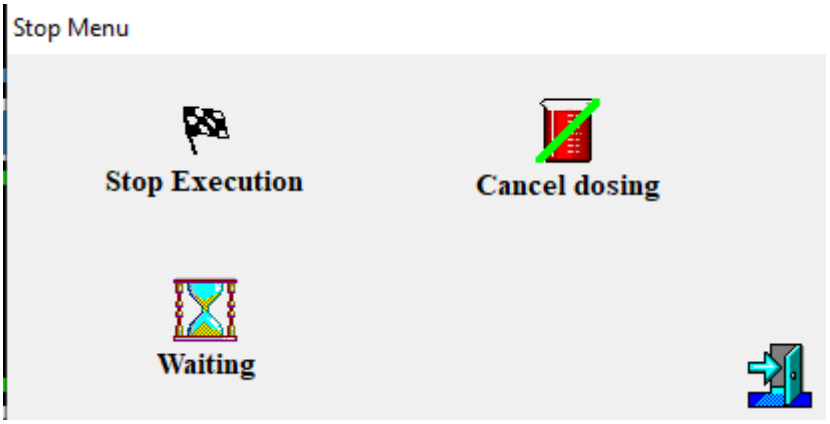


Figura 3.7: Menú 5. Menú de Parada

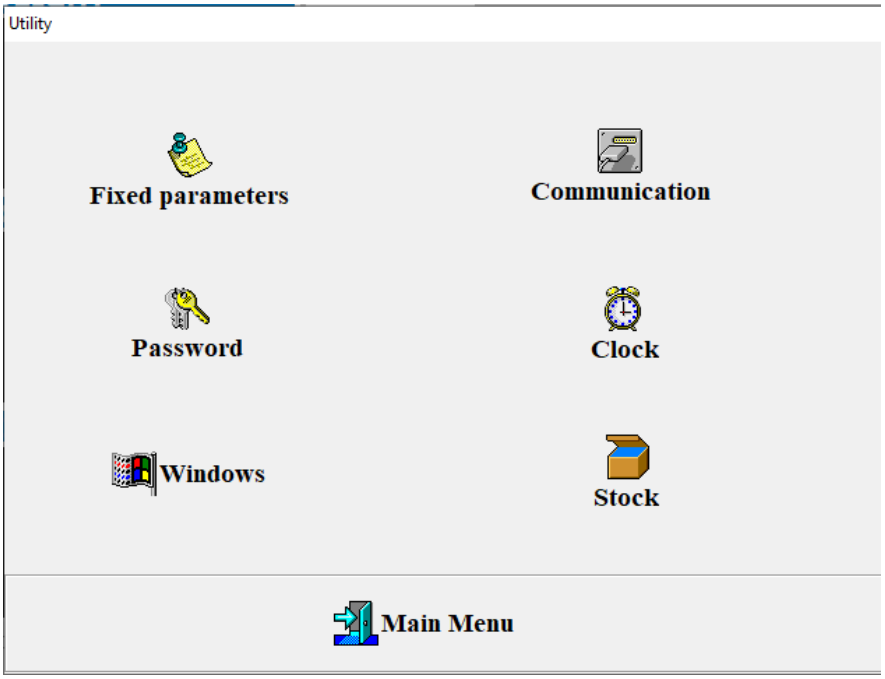


Figura 3.8: Menú 6. Menú Utilidades

  - Control de carro.
  - Control de código de barras.
  - Impresora: Presenza da impresora.

  Os parámetros fixos son modificables mediante o botón *Modificar* do formulario. Unha vez modificados, úsase o botón *Envía* para transferilos á estación de pesado.

- *Comunicación*: Permite visualiza-lo estado da conexión entre o PC e o PLC de control da máquina. No noso caso non aplicaría.

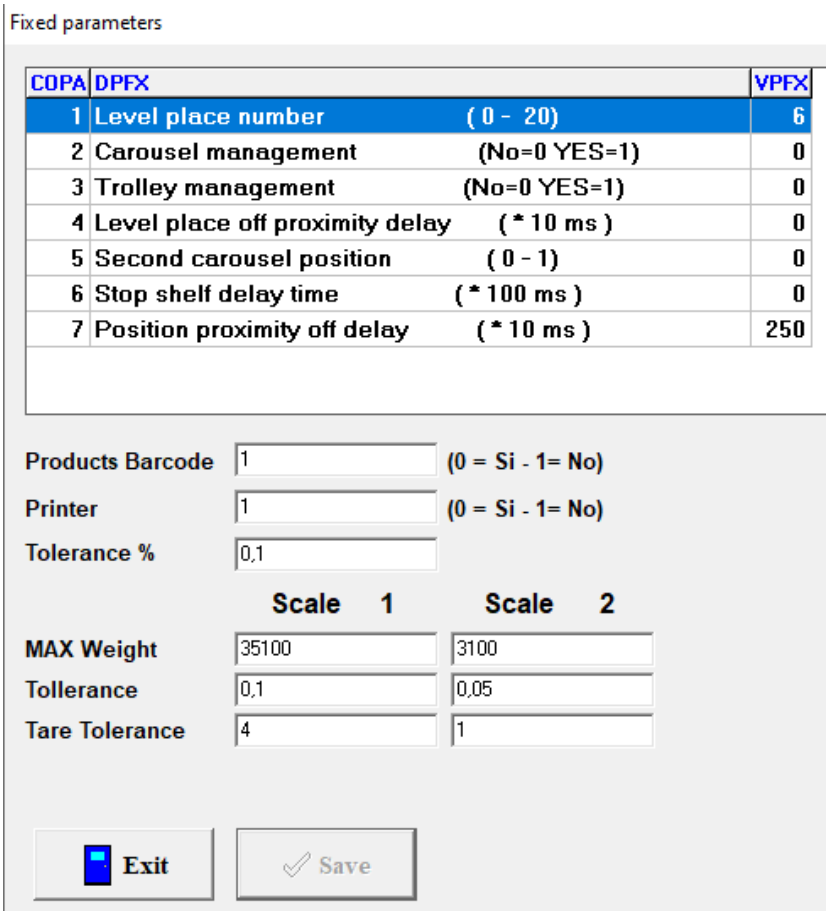


Figura 3.9: Menú 6.1. Parámetros Fixos

- *Reloxo*: Actualizar data e hora.
- *Contrasinal*: Inserir e modificar contrasinais. Orientado a establecer distintas claves en función do posto do operario. Así, o sistema pode solicitar claves distintas para modificar produtos de almacén, saír ao sistema operativo, etc.
- *Xestión de almacén.* Permite a xestión dos produtos, codificación e cantidades, incluído un control de aviso para produtos por baixo dunha cantidade mínima.
  - Código: Hai que insistir en que ten que se-lo código definido a nivel corporativo.
  - Descrición. O nome do colorante.
  - Andel-Posición: A localización da caixa activa. Non coincide exactamente co sistema de mapeo de Lavamar, que inclúe como prefixo un código de zona, é dicir: `zona.columna.fila`.
  - Data de carga, que é cando se repuxo colorante.

- Cantidade cargada.
- Stock mínimo. Pódese definir un stock mínimo que o sistema usa para amosalos colorantes con reservas baixas mediante un botón.
- Stock actual. En base o que se carga, vaise descontando cada pesada, obtendo o remanente. En Lavamar, o control de stock, faise dun xeito distinto, polo que isto non aplicaría.

Unha vez modificados hai que envía-lo estado do almacén á estación de pesado.

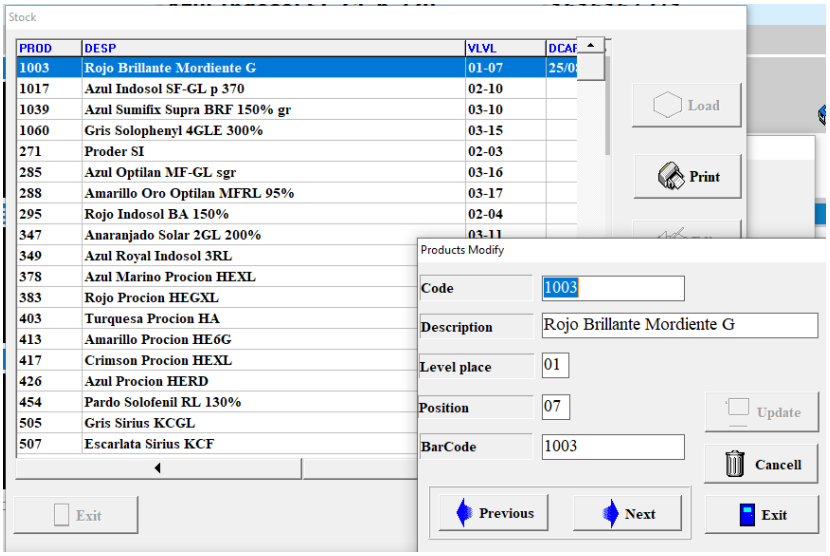


Figura 3.10: Menú 6.2. Xestión de Almacén

Windows Permite pecha-lo programa e saír ao sistema operativo. Esixe a inserción dunha clave a modo de seguridade.

3.3 Extracción da información

Non hai código fonte dispoñible, polo que esta segunda fase habitual no proceso de análise do sistema debe realizarse de xeito indirecto.

3.3.1 Análise da interface do programa

O obxectivo deste proxecto é que o programa funcione e sexa integrado no sistema produtivo. Pero existe a posibilidade de que non sexa así, polo que haberá que facer un novo programa que emule o funcionamento deste e que se pareza ó amosado na imaxe 3.2. Vemos que o obxectivo é ir pesando até unha cantidade que sexa validada polo programa e entre dentro dos limiares de tolerancia.

3.3.2 Ficheiros de configuración

Este programa de pesaxe usa ficheiros de texto plano como modo para modifica-lo seu comportamento e mailas súas capacidades. Coñecer ben a estrutura, os termos clave e os valores correctos que deben indicarse, é fundamental para que o sistema cumpra co que se espera del. Hai ademais dous tipos de ficheiros, que se dirixen a ámbitos distintos:

1. Propios das balanzas activas: Estes ficheiros encárganse de representa-la configuración das distintas balanzas que poden ser usadas polo programa. Un ficheiro por cada tipo de balanza.

2. Propios do sistema: Nestes ficheiros, indícanse distintas configuracións para que a aplicación funcione: portos serie, enderezos IP,...

3.3.3 Ficheiros de log

A aplicación permite a activación dun *log* en vivo das respostas da balanza e tamén xera *log*s de alarmas e de funcionamento en ficheiros de texto.

3.3.4 Ficheiros de importación

Durante a busca de información, atopamos que o sistema permitía a compartición de información con sistemas externos mediante tres tipos de ficheiros de texto formateados, cada un cunha finalidade distinta:

- Ficheiro de texto `magazz.tmp`: Permite a inserción de tricomías dentro do programa.
- Ficheiro de texto `prodotti.tmp`: Permite a inserción de colorantes no almacén de colorantes do programa de pesaxe e información da súa localización.
- Ficheiro de texto `pesate.tmp`: O programa vai xerando este ficheiro cada vez que conclúe a pesaxe dunha tricomía, para que sexa procesado por un programa externo. O procesamento deste ficheiro non formará parte da funcionalidade do sistema.

A información relativa á estrutura destes ficheiros de texto atópase no anexo B.

3.4 Identificación de fontes de datos

Debemos identificar detalladamente as fontes de datos para saber onde podemos obte-la información necesaria, e que fontes son consumidoras e cales produtoras de información, para definir claramente o fluxo dos datos no sistema. Así, a fonte de datos principal, na que se garda e xera información e a Base de datos Paradox situada no directorio da aplicación

`./Data`. Nesta base de datos recíbense as tricomías a pesar, mais tamén se almacena o estado da pesaxe, cantidades, horas de inicio e fin, ...O acceso á dita base de datos é local para o programa de pesaxe, pero será remota para o resto de elementos do sistema. Dita conexión de forma remota ás fontes de datos, será mediante protocolo Server Message Block (SMB). Unha fonte de datos fundamental tamén, serán as lecturas que forneza a balanza seleccionada para face-las probas do noso proxecto.

A listaxe completa dos campos das táboas implicadas na execución do proxecto están dispoñibles con máis detalle no anexo C. A continuación nomeamo-las táboas desta base de datos e os seus campos máis importantes.

- RICHDOSA: Almacena a cabeceira da tricomía para pesar.
  Campos salientables:
  - COMM: O nº de tarefa. En Lavamar, idMaquinada.
  - NMAC: O nº de máquina onde se efectúa a tintura.
  - FASE: O momento no que a tricomía debe engadirse na máquina. COMM, NMAC e FASE forman a clave primaria para identificar unha tricomía.
  - INIZ: Indica día e hora de inicio da pesaxe da tricomía.
  - FINE: Indica día e hora de fin de pesaxe da tricomía.
- INGDOSA: Almacena os colorantes da tricomía e a cantidade para pesar.
  Campos salientables:
  - COMM: O nº de tarefa. En Lavamar idMaquinada.
  - NMAC: O nº de máquina onde se efectúa a tintura.
  - FASE: O momento no que a tricomía debe engadirse na máquina.
  - QUANT: Indica a cantidade a pesar do colorante.
  - NUMP: Indica a orde na que debe ser pesado o colorante. Este campo e tódolos anteriores forman a clave primaria para identificar un produto dentro dunha tricomía. É unha boa idea porque permite a división dun mesmo produto dentro da tricomía se as básculas estivesen por riba do seu límite de pesado.
  - CODICE: O código do produto. Dito produto debe figurar no almacén do programa de pesaxe (táboa STOCK).
  - DESC: O nome do colorante
  - BARCODE: O código de barras para identifica-lo produto. Pode ser igual que CODICE, ou ben usa-lo código do fornecedor. No noso caso será igual á CODICE, porque na gran maioría dos fornecedores, as caixas non traen o número de produto

en forma de código de barras, polo que deixa de ser útil. O noso xestor de almacén asignará unha etiqueta co código de barras correcto á caixa activa, no caso que queiramos usar esta funcionalidade.

- STOCK: Almacén onde figuran tódolos colorantes susceptibles de seren pesados e a súa colocación. Tamén se podería usar a modo de inventarío porque permite o almacenamento da cantidade actual dun produto modificándoo con cada pesada. Tamén avisa de stock mínimo.
  Campos salientables
  - PROD: O nº do colorante. Debe coincidir co identificador idMaterial que se usa en Lavamar
  - DESP: O nome do colorante.
  - VLVL: A localización do colorante (fila-columna).
- ALARMI: A listaxe de alarmas.
- STATODOSAGGIO: O estado da pesada actual. Non tocaremos está táboa. Nela o sistema de pesaxe vai inserindo información só da tricomía activa, almacenando o estado no que se atopa.

Dada a ausencia de integridade referencial entre estas táboas, non se inclúe neste punto ningún diagrama.

3.4.1 Produtores de información

Especificamos agora cales serían as formas de achegar información ó sistema mediante a inserción de datos. Dúas vías:

- Mediante inserción directa nas táboas, que permite unha rápida dispoñibilidade por ser unha operación síncrona. Operacións básicas (INSERT, SELECT, DELETE, UPDATE) mediante SQL.
  Problemática:
  - Punto crítico de rendemento. Non destaca por ser rápida a BD Paradox.
  - Seguridade de acceso, ó depender do protocolo SAMBA.
  - Dispoñibilidae de acceso ás táboas inherentes á LAN corporativa.
  - Problemas de bloqueos ó compartir a base de datos co programa de pesaxe. Corrupción de índices.

- Mediante ficheiros de texto codificados procesados polo programa nun directorio circular. Transferencia de ficheiros
  Problemática:
  - Información representada por unha estrutura estritaB.
  - Política de nomes e localización. Tempos que seguí-la nomenclatura indicada e almacena-lo ficheiro no directorio correcto.
  - Políticas de creación e borrado dos ficheiros. Hai que saber que facer co ficheiro unha vez procesado
  - Problemas de interbloqueo. Posíbilidade de bloqueo na creación ou borrado dos ficheiros.
  - Seguridade de acceso. Protocolo Server Message Block (SMB). Os ficheiros tamén serían compartidos baixo este protocolo.

Ademais, xa dentro da mesma execución do programa, teríamos as lecturas producidas pola báscula, que se está calibrada é fácil verifica-la súa fiabilidade.

3.4.2 Consumidores de información

Os consumidores da información serán o propio programa de pesaxe e o programa de planificación corporativo.

- O programa de pesaxe: Precisa das tricomías e do estado do almacén para poder executa-las accións requiridas.
- O programa de planificación: Pode consultar mediante a interface que se creará se unha tricomía finalizou.

3.5 Situación actual: Arquitectura

Grazas os diagramas C4 [5] podemos apreciar a arquitectura actual do sistema de Lavamar e como funcionaría o sistema de pesaxe se estivese funcionando tal e como estaba pensado. As figuras 3.11, 3.12 e 3.13 presentan respectivamente os niveis de contexto, contedor e compoñente desta notación.

Como se aprecia, o sistema actual pivota ó redor do sistema de Termoelettronica, concretamente no sistema Termorecipe, que é quen se encarga da xestión dos distintos produtos químico e a súa dosificación se fose o caso. O problema é que se o usásemos como está pensado, teríamos que inserir a man as tricomías de tódalas ordes de traballo que van a máquinas

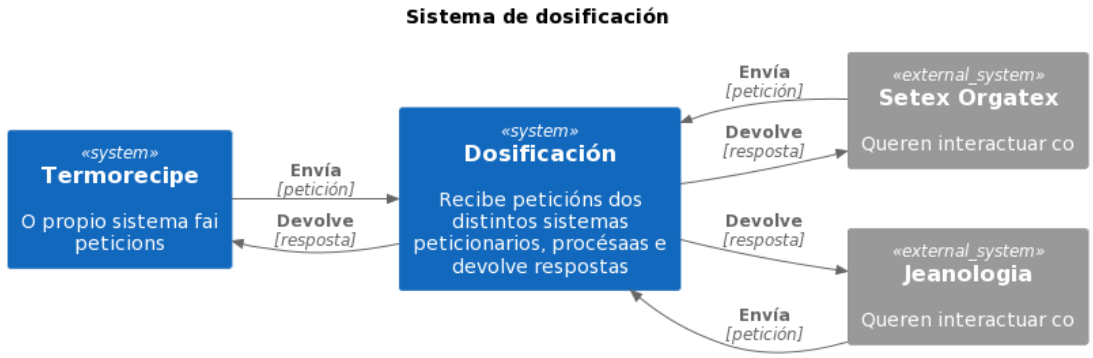


Figura 3.11: C4. Contexto situación inicial

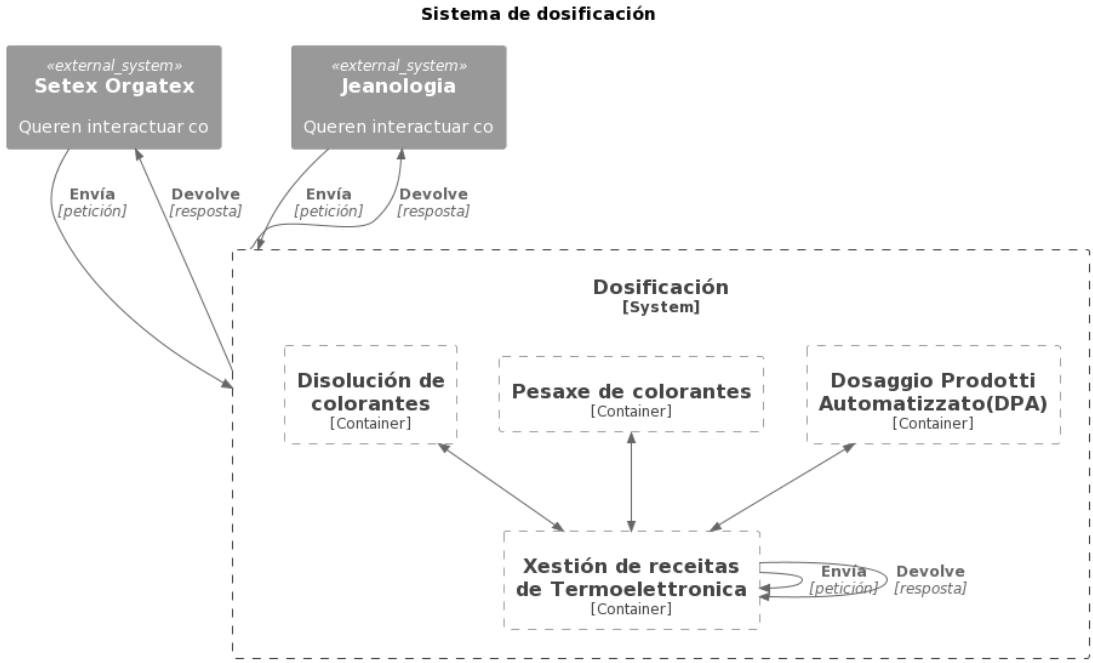


Figura 3.12: C4. Contedor situación inicial

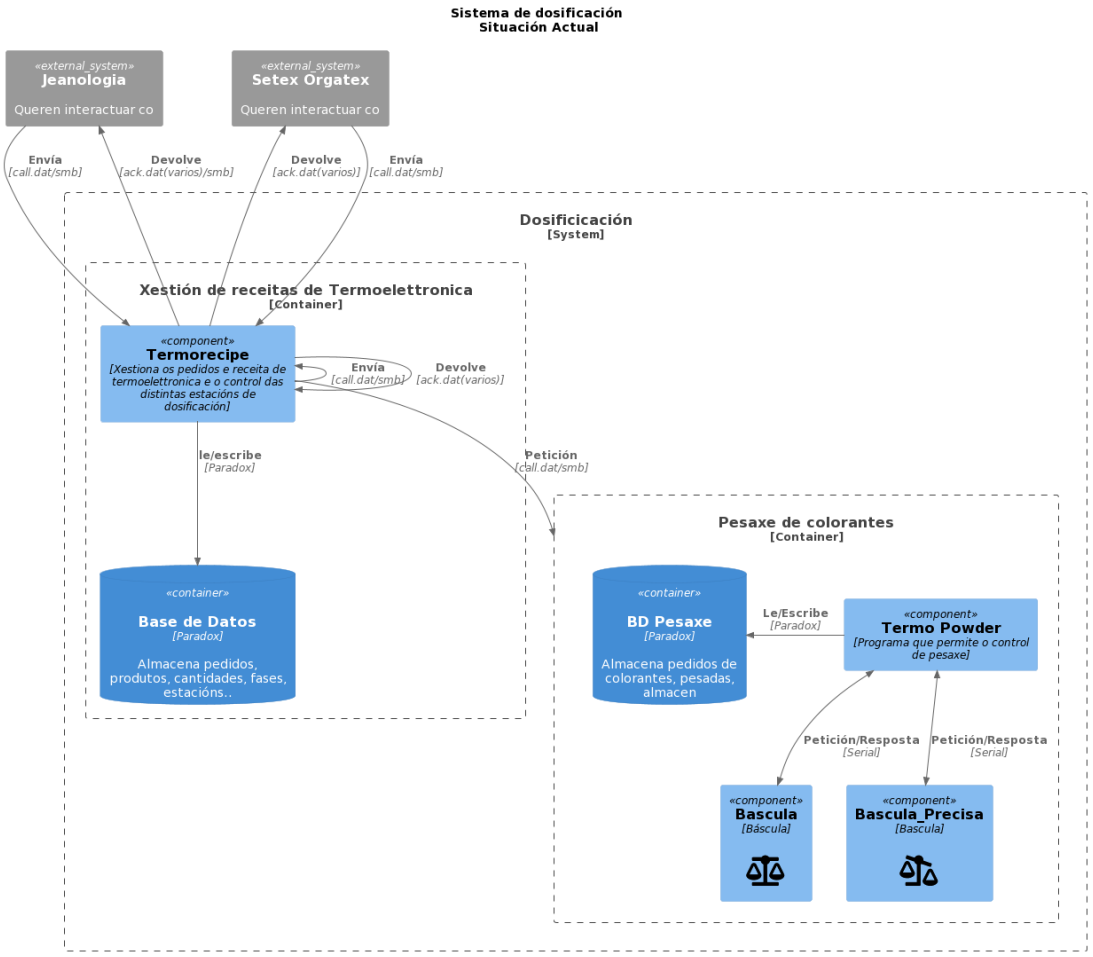


Figura 3.13: C4. Compoñente situación inicial

que traballan con Termoelettronica, e tamén das máquinas de Jeanología. Setex, o outro fabricante, ten unha interface para solicitar produtos (supostamente incluídos colorantes), pero non compensa probalo en vista do comentado. Queremos que o sistema sexa fiable no intercambio de información, polo que hai que evitar a inserción de datos de forma manual porque iso inevitablemente leva a aparición do erro humano.

3.6 Conclusión

En vista de toda a información analizada e dunha comprensión profunda da estrutura inicial do sistema de partida, das deficiencias identificadas tanto no estado actual, como se o sistema funcionase integrado na forma na que estaba pensado tal e como amosan os diagramas C4 do punto anterior, estase en disposición de formular unha hipótese que sirva de aproximación inicial cara a reenxeñaría do sistema cunha arquitectura tentativa e a análise dos requirimentos.

Esta fase de análise do sistema é crucial para o éxito da reenxeñaría, pois establece as bases sólidas sobre as cales se construirá todo o proceso de reformulación. A visión inicial transfórmase nunha estratexia concreta e fundamentada para a mellora do sistema existente.

Capítulo 4

Análise de requirimentos

O obxectivo principal deste capítulo é proporcionar unha visión completa e detallada dos requisitos do sistema. Iso inclúe tanto os requisitos funcionais, que describen as accións e funcionalidades que o sistema debe realizar, como os requisitos non funcionais, que se refiren a características de rendemento, seguridade, usabilidade e outros aspectos que afectan á calidade global do sistema.

4.1 Requirimentos Funcionais

Abordamos aquí os requisitos funcionais do noso sistema. A decisión de agrupa-los requisitos en tres bloques distintos está fundamentada na necesidade de organización e claridade na presentación dos mesmos. Cada bloque representa unha área funcional diferente do sistema, o que permite unha visión máis específica e detallada das funcionalidades asociadas á cada área. Ademais, esta abordaxe axuda a que no desenvolvemento se poidan identificar facilmente os requisitos que son relevantes en cada subsistema.

4.1.1 Referidos ó programa de pesaxe

Lista dos requisitos esixidos ó programa de pesaxe, ben sexa resultado da recuperación do sistema TermoPowderM, ben sexa pola creación dun novo programa de pesaxe implementado dende cero se TermoPowderM fose irrecuperable.

- Xestión de ordes de traballo en forma de tricomías para seren pesadas.
- Procesamento guiado de ditas tricomías, axustándose ao seguinte fluxo:
    1. Selección da tricomía a pesar .
    2. Información en pantalla do n-colorante para pesar.
    3. Información de báscula preparada.

4. Colocar bandexa na báscula.
5. Tarado da balanza.
6. Inicio da pesaxe até acadar unha cantidade que entre dentro do limiar de tolerancia da balanza.
7. Fin de pesaxe do colorante.
8. Continuar co colorante n+1 da tricomía se o hai e volver ó punto 1.

- Xestión de alarmas: visualización e borrado.
- Xestión de produtos no almacén: creación, edición e borrado de colorantes, coa seguinte información relevante:
  - Código do colorante.
  - Nome do colorante.
  - Localización do colorante.
- Creación de tricomías de forma directa
- Histórico de pesadas
- Anulación de pesada (tricomía ou colorante activo)

4.1.2 Referidos ó programa de planificación implantado en fábrica

O módulo de planificación de Lavamar é o programa de xestiona-las ordes de traballo sobre a roupa dispoñible para distribuí-la polas máquinas axeitadas usando as Receitas de produción correspondentes a cada caso. A súa funcionalidade principal é fornece-las tricomías que deben ser pesadas.

- Exportación baixo demanda da información referida a tódalas tricomías presentes nunha Receita planificada cara un documento XML ou nun XML dentro dun *string*
- Exportación automática cada vez que se planifica algo.

4.1.3 Referidos o programa interface que se comunicará coa base de datos de TermoPowderM e coa contorna corporativa

Este módulo é o intermediario entre o sistema corporativo e o sistema de pesaxe, polo que o cumprimento estrito dos seus requisitos funcionais é imperativo para unha completa integración entre tódolos módulos.

- Engadir tricomías ou cun ficheiro XML ou cunha cadea XML en TermoPowderM[1].
- Borrado de tricomías mediante un ficheiro XML ou unha cadea XML en TermoPowderM.
- Borrado de produtos do almacén de TermoPowderM.
- Listaxe de tricomías activas en TermoPowderM.
- Cargar colorantes do almacén de Lavamar na base de datos de TermoPowderM, con toda a información relevante.
- Devolve-lo estado de pesaxe da tricomia.

4.2 Requirimentos Non Funcionais

Os requisitos non funcionais abordan criterios importantes para o seu rendemento, eficacia e fiabilidade do sistema. Estes requisitos inflúen na experiencia global do usuario e no desempeño do software, xa que definimos parámetros críticos que afectarán a forma na que se comporta o sistema, a capacidade para manter unha carga de traballo variada ou a súa resistencia a fallos ou ataques, e que tamén se satisfagan as necesidades globais da organización e dos usuarios finais.

- Facilidade de uso:
  - TermoPowderM: Como a execución da pesada é guiada é doada de seguir.
  - A integración no sistema de planificación será transparente ó operario.
- Robusto. Evitar corrupción na base de datos. A corrupción de índices en táboas Paradox conectadas en rede ten unha alta incidencia, baseándonos na experiencia previa con outros sistemas de Termoelettronica da empresa.
- Portable. Debe funcionar con distintas versións de Windows e entornas heteroxéneas.
- Rendemento. Os datos, unha vez emitidos polo sistema de planificación, deben estar dispoñibles no sistema de pesaxe en menos de 5 segundos no caso de inserción directa en táboas. No caso de uso de ficheiros, a interface debe xerar o ficheiro de intercambio e copialo en destino en 5 segundo como máximo.
- Seguridade:

[1] Cando dicimos TermoPowderM, referímonos ó programa de pesaxe, pois pode ocorrer que teñamos que desenvolver un partindo de cero.

  - Sistema distribuído. Seguridade herdada da configuración da rede LAN da empresa.
  - Programa de pesaxe. Roles autorizados ás distintas funcionalidades.

- Dispoñibilidade:
  - Acceso directo á base de datos do programa de pesada para a súa modificación e lectura.
  - Acceso directo á localización do programa para o uso de ficheiros temporais de intercambio de información.

- Fiabilidade: O sistema debe funcionar correctamente, intercambiando información veraz e precisa. As básculas deben estar calibradas, pois a súa lectura tamén debe ser precisa e verificable.

- Recursos:
  - Ordenador con pantalla táctil.
  - Lector con código de barras.
  - Sistema operativo Windows XP ou superior.
  - Acceso á rede Local Area Network (LAN) da empresa.
  - Básculas compatibles co sistema de comunicación SICS [6].

- Soporte para múltiples linguas.

Capítulo 5

Deseño

Nesta sección, abordaremos a fase de deseño do noso proxecto. É un paso crítico no proceso de desenvolvemento, xa que establece as bases para a implementación do sistema e inflúe directamente na súa eficiencia, rendemento e usabilidade. Nesta etapa, traduciremos os obxectivos e requisitos previamente identificados na fase de análise nunha arquitectura detallada e nun plan de implementación concreto.

5.1 Fusión de vistas

Grazas o visto na fase de análise (capítulo 3) temos que mellorar ou transformar a arquitectura existente noutra delimitada polos novos requirimentos e evita-los problemas da arquitectura actual. O sistema orixinario estaba pensado para traballar de forma dependente do sistema de dosificación de produtos químicos de Termoelettronica, que é un sistema extremadamente antigo. Dito sistema sería o encargado de envía-las tricomías ó programa de pesaxe, pero iso suporía a inserción a man dos colorantes e as súas cantidades, co que perderíamos seguridade e fiabilidade no intercambio de información, a trazabilidade veríase comprometida e inevitablemente aparecería o fallo humano. Ademáis a comunicación entre ámbolos dous sistemas é mediante ficheiros de texto procesados a través de directorios de monitoreo continuo, o que non é unha forma óptima de intercambio de datos, xa que é complicado controlar se os datos xa foron procesados de xeito correcto.

Co coñecemento extraído nos puntos anteriores, formularemos unha proposta arquitectura tentativa.

5.1.1 Reconstrución do sistema

Nesta etapa, abordarémo-la transformación do sistema para garantir que atenda aos novos requisitos e estándares establecidos. Ó longo desta sección, exploraremos en detalle os pasos necesarios para recrear e optimiza-lo sistema.

Son elementos a reutilizar:

- O propio software do que queremos facer reenxeñaría.
- De non funcionar, crearemos un programa para efectuar pesadas.
- Base de datos do programa, se aquel funcionase.
- Hardware reutilizable: ordenador táctil, lector de código de barras, balanza de reposto.

A continuación, detallámo-los pasos para coñecer un estado inicial dunha potencial reconstrución:

- Instalación do programa TermoPowderM no ordenador táctil.
- Instalación do software necesario para que funcione o programa: Borland Database Engine (BDE).
- Conexión á rede LAN.
- Conexión e comunicación coa(s) balanza(s) que efecturá a pesaxe. Dita báscula é unha Mettler-Toledo modelo Mettler BBK 422 - 3 XS[7], que permite un máximo de 3100 gramos cunha tolerancia de 0,01g.

Unha vez feito isto, executarase a aplicación para confirma-lo seu funcionamento.

5.1.2 Problemáticas esperables

Ante unha primeira execución da aplicación, os fallos máis probables son:

- Fallo de execución do programa por mala configuración ou polo uso de bases de datos Paradox.
- Fallo na comunicación coa báscula.
- Fallos en como debe funcionar o programa.

5.1.3 Plan de continxencia

A lista de actuacións previstas ante as problemáticas relatadas é a seguinte:

- Revisión e instalación das librarías coa versión axeitada para o correcto funcionamento do programa. Nomeadamente Borland Database Engine (BDE) [8].
- Verifica-los ficheiros de configuración das básculas que se encontran no directorio principal do programa.

- Proba/erro iterativamente até conseguir que o fluxo de traballo sexa o correcto.
- Crear un programa para a diagnose da comunicación coa báscula.

5.1.4 Toma de decisións

Finalmente, en base o resultado das distintas actuacións do plan de continxencia, podemos definir varios pasos a seguir:

- Se non funciona o programa, ben porque non arrinca, ben porque o seu funcionamento é inadecuado, crear de cero un programa que cumpra cos requirimentos (recollidos no capítulo 4).
- Se non hai conexión coa báscula:
  - Revisa-lo ficheiro de configuración activo para atopar fallos que impidan dita conexión.
  - Se non hai éxito cos ficheiros de configuración, crear un programa de diagnose para identifica-las respostas da balanza ás distintas ordes e poder trasladar ese coñecemento para crear un ficheiro de configuración operativo.
- Se se crea un programa de diagnose, volver repeti-las probas coa información que proporcione dito programa.
- Se co programa de diagnose non hai avances ou coa conexión coa báscula, ou co funcionamento xeral de TermoPowderM, crear un programa novo para efectua-la pesaxe que cumpra os requirimentos esixidos(4).

5.1.5 Arquitectura tentativa

Presentamos a continuación os diagramas C4 do sistema proposto, que poden examinarse nas figuras 5.1, 5.2 e 5.3.

O que debemos reparar neles é que nesta arquitectura tentativa aparece unha interface que comunicará o sistema corporativo co sistema de pesaxe de forma directa, eliminando completamente a dependencia co sistema de dosificación de Termoelettronica[9]. Con isto queremos conseguir unha comunicación fiable, controlada e baixo demanda do sistema corporativo co sistema de pesaxe, evitando a interferencia de sistemas de terceiros. Ó estar ante un escenario tan heteroxéneo, onde conviven tantos sistemas distintos, é fundamental que o único sistema que realmente xestiona o fluxo produtivo da empresa sexa quen teña o control sobre os datos que fornecen neste caso ó programa de pesaxe.

Vemos como se aprecia este cambio fundamental no diagrama de contexto (figura 5.1). O que interesa é te-lo control do programa de pesaxe sen que sistemas externos interfiran no seu funcionamento.

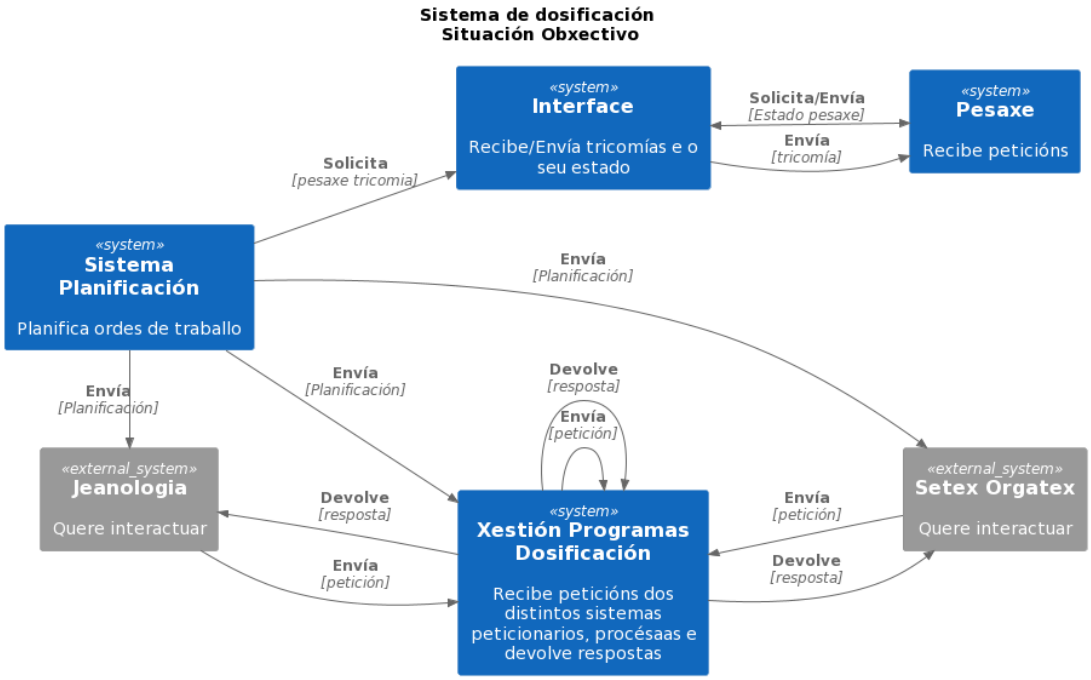


Figura 5.1: C4. Contexto situación obxectivo

A idea é que o Sistema de planificación envíe cara a interface as tricomías que se queren pesar, con toda a información necesaria para desenvolver dita tarefa. A interface, procesará ese conxunto de tricomías e inserirá dita información no sistema de pesaxe, para nutri-la lista de tricomías a pesar que estarán dispoñibles no programa de pesaxe.

No seguinte diagrama, o de contedores (figura 5.2), vemos que a interface entre o Celme[1] e o sistema de pesaxe, é moi especializado. A súa función principal será transferir ó sistema de pesaxe tricomías que deben ser pesadas, solicitadas baixo demanda en base as planificacións que se van xerando. Finalmente, o diagrama de compoñentes (figura 5.3) detalla aínda máis a situación obxectivo.

5.2 Deseño detallado e plan de implementación

Nesta sección imos entrar nos detalles máis íntimos da arquitectura, definindo compoñentes e as súas interaccións, e trazar un plan claro e organizado para a implementación co

[1] Nome que lle da o departamento de sistema ó sistema corporativo

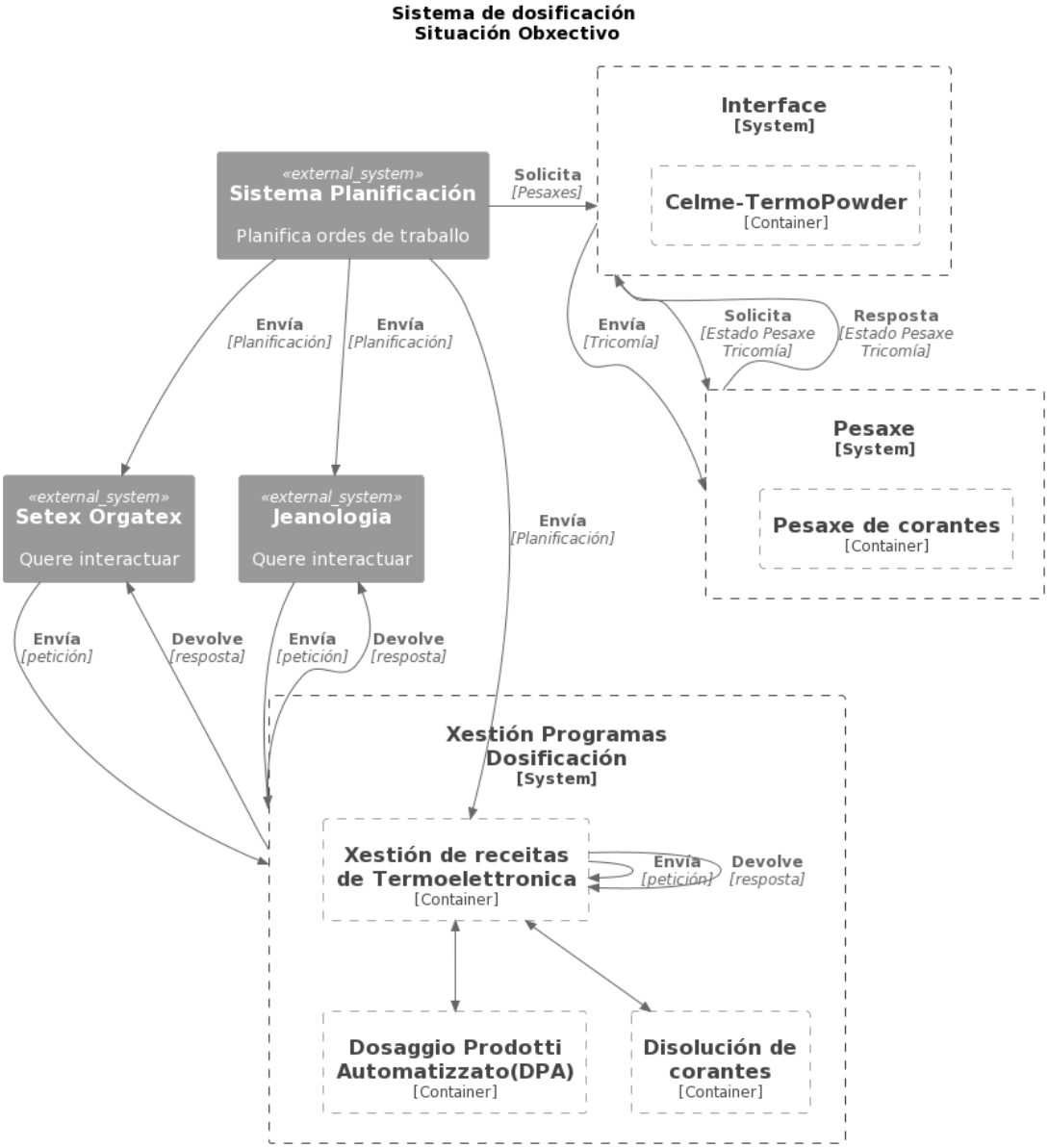


Figura 5.2: C4. Contedor situación obxectivo

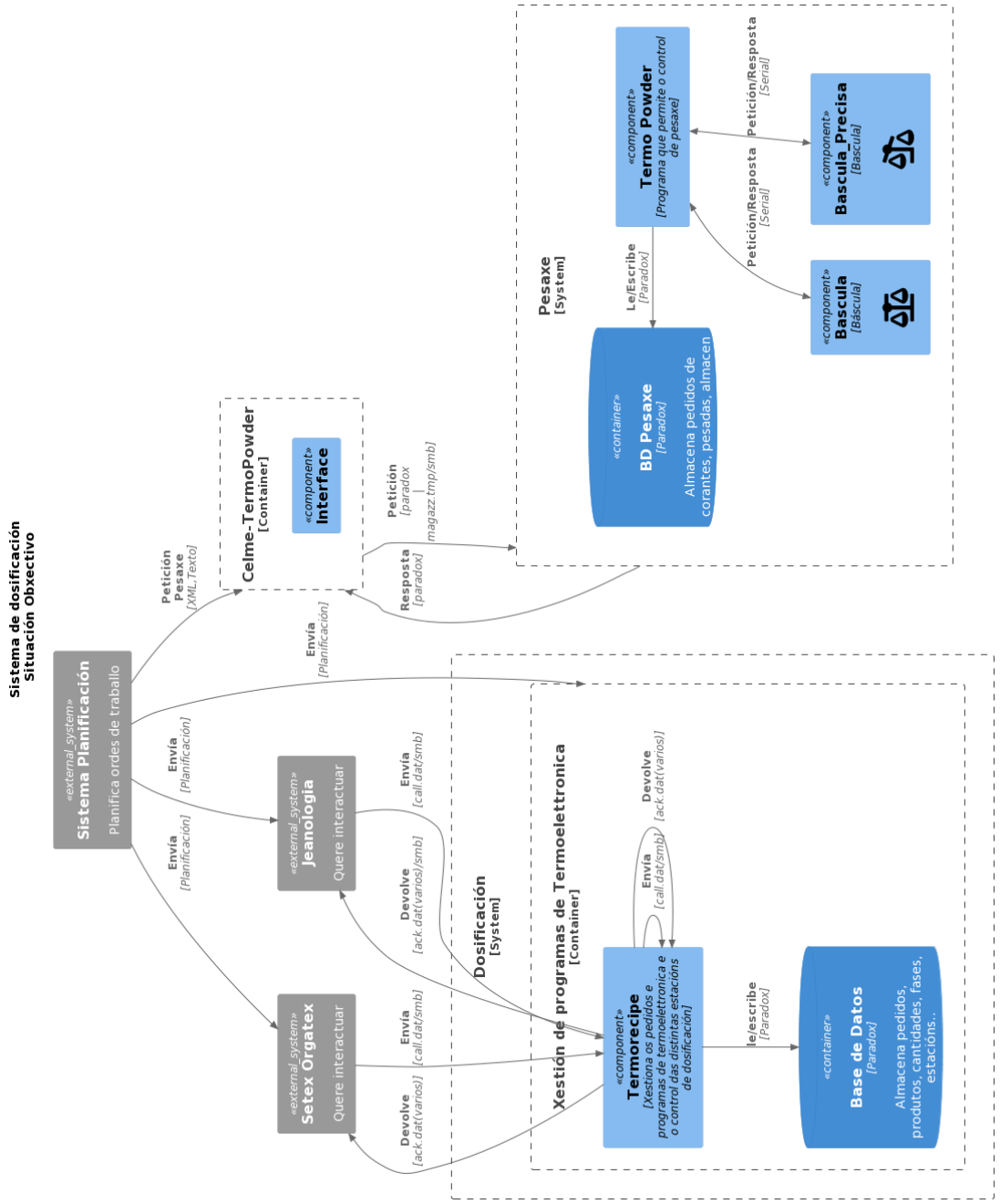


Figura 5.3: C4. Compoñente situación obxectivo

fin de lograr eficiencia e cohesión do sistema no seu conxunto. Todo o recollido previamente transfórmase nun plan tanxible para acadar culminar con éxito o proxecto.

1. Programa de diagnose.
   Crear un programa de diagnose se hai problemas coa comunicación coa impresora.

   - Uso de C# para o desenvolvemento.
   - Implementación dun programa de diagnose para comproba-la comunicación coa báscula candidata seguindo o protocolo SICS[2]
   - Analiza-las respostas ás ordes de inicialización, tarado, pesado continuo, pesado puntual e outras para poder conformar máis adiante os ficheiros de configuración
   - Implementar un programa de consola que reciba instrucións mediante interacción por liña de comandos.

2. TermoPowderM
   Co sistema de pesaxe debemos centrarnos sobre todo nos ficheiros de configuración para avaliar que a comunicación coa báscula funciona, e tamén que se realiza correctamente o fluxo de pesaxe.

   - Creación dun ficheiro de configuración específico para a báscula coa que efectuaremolas pesadas baséandonos tanto nos ficheiros existentes (de varias marcas: AND, Bottaro, Sartorius, Mettler,...) e os resultados obtidos co programa de diagnose. Dito ficheiro de configuración levará o nome de `Scale.Mettler.PFG.ini`
   - Asignación dun porto serie para o uso da báscula, e deixar outro reservado para no futuro incorporar outra báscula se é posible. O programa TermoPowderM permite até dúas balanzas simultáneas. De inicio, o COM1 e o COM2 serán os portos asignados ás dúas potenciais balanzas conectadas. Os demais parámetros relacionados coa comunicación serie (baudios, paridade, control de erros,...) serán definidos na fase de implementación.
   - Probas de pesaxe cunha tricomía creada manualmente *ad hoc*. O sistema debe realizar o fluxo de traballo completo indicado nos requirimentos (páxina 33). O programa de pesaxe, unha vez conclúe a pesaxe dunha tricomía completa, non a elimina do rexistro de tricomías. Simplemente indica que se completou poñéndoo a fonte a negro.
   - Utilización optativa dun lector de código de barras para escanea-los colorantes.

[2] https://www.geass.com/wp-content/uploads/filebase/Utilita/MT-SICS.pdf

Non se van empregar tódalas táboas da base de datos. Só as imprescindibles para que se cumpran os requisitos. Así, as táboas da base de datos Paradox de TermoPowderM coas que se vai interactúar(C) son:

- Táboa RICHDOSA: Onde se almacena a información da tricomía a pesar.
- Táboa INGDOSA: Onde se almacena o detalle da tricomía a pesar.
- Táboa STOCK: Onde se garda a información sobre os colorantes dispoñibles para seren usados na tricomía.

3. Adaptación do programa de planificación

   - Modificación do programa de planificación para que cando se efectúe unha planificación cara unha máquina SETEX, o sistema obteña da Receita que se vai executar, as tricomías que conteña (normalmente unha), e devólvaa en forma de ficheiro XML ou de cadea con datos XML.
   - O uso de planificacións cara a máquinas con SETEX[10] débese a que esas Receitas, mediante unha estrutura filter & pipe (representada na figura 5.4), producen unha Receita final cos datos completamente calculados. O impacto sobre o sistema é moi pequena pois so ten que procesar un XML dun tamaño aproximado de 15-20k, que é a Receita completamente serializada que recolle á saída do último filtro.
   - Permiti-lo envío dunha tricomía dun proceso xa planificado baixo demanda.

4. Interface con TermoPowderM
   As pautas a seguir co interface que conectará o sistema corporativo co sistema de pesaxe son:

   - Uso de C# para o desenvolvemento.
   - Crear unha aplicación de consola que permita a recepción de distintas ordes por liña de comandos.
   - Por liña de comandos debe recibir tamén a localización do ficheiro XML ou ben unha cadea cun XML ben formado para ser procesada.
   - Comezar por funcionalidades que axuden a verifica-la conectividade coa base de datos Paradox do programa de pesaxe (páxina 34), tendo en conta que é unha base de datos remota.
   - Deberá haber unha clase que englobe as interaccións coa base de datos Paradox.
   - Deberá haber unha clase que englobe as interaccións coa base de datos corporativa. Realizará accións de consulta sobre os colorantes activos presentes en fábrica.

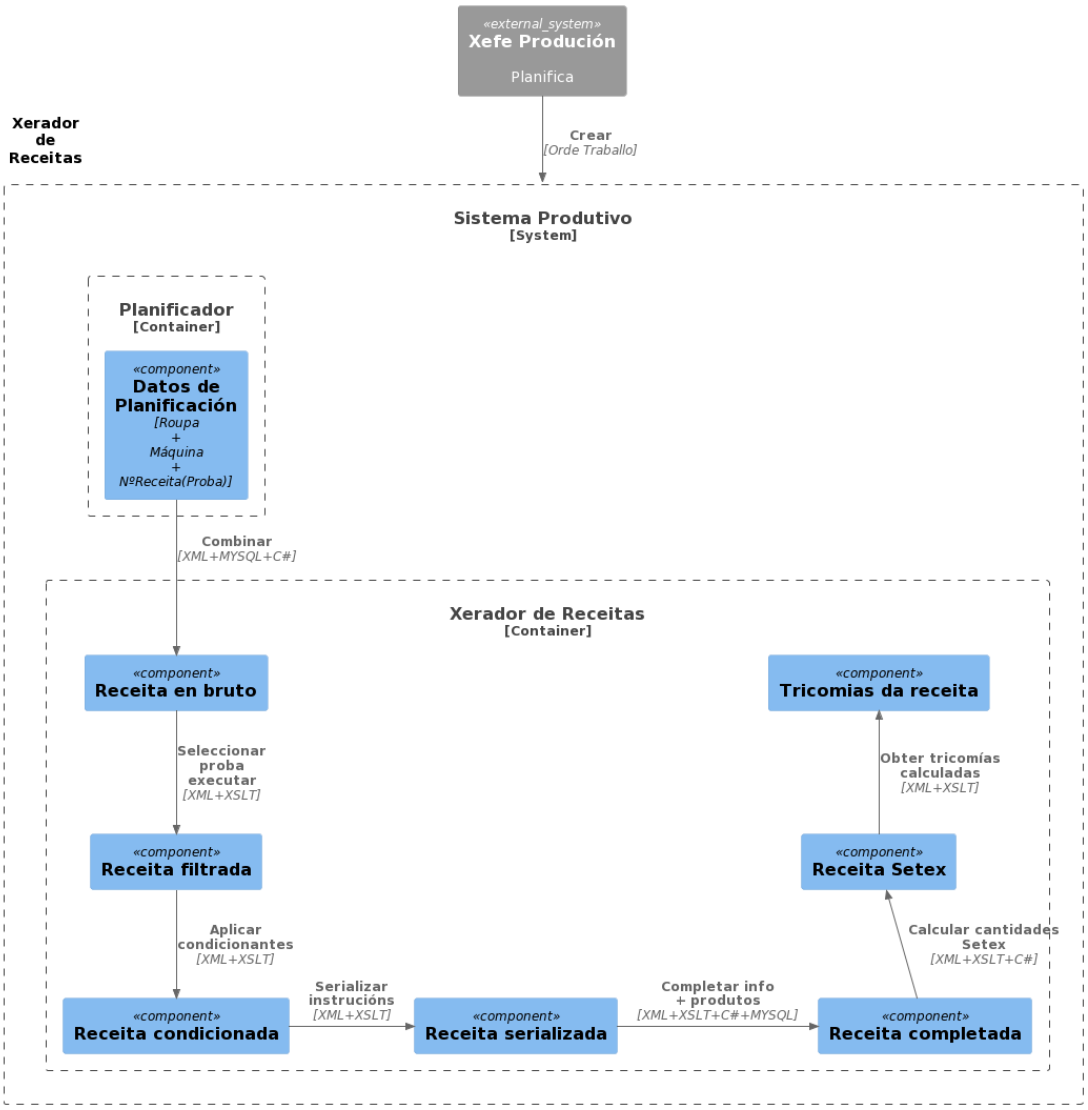


Figura 5.4: Anexo de filtro para obtención de tricomía

- Debemos inserir na táboa RICHDOSA a información sobre a tricomía e na táboa INGDOSA a información sobre os colorantes que a compoñen.
- Debemos inserir na táboa STOCK os colorantes activos que imos usar nas pesaxes, pois senón o programa de TermoPowderM non permite continúa-la operación.

5.2.1 Tecnoloxías e ferramentas

Nesta sección explicaremo-las distintas tecnoloxías que usaremos para implementa-lo sistema.

Usaremos de forma xeral C# pois é a linguaxe que se está a usar para os novos desen-

volvementos da empresa, polo que é unha ferramenta coñecida e non precisa de tempos de aprendizaxe. Así tanto o programa de diagnose (no caso de implementarse) coma a interface de conexión co programa de pesaxe faranse nesta linguaxe mediante o uso do IDE Visual Studio[11].

A parte da adaptación do programa de planificación usarase Visual Basic[12], pois é na linguaxe que está escrita esa parte e fará uso das librarías de XML de Microsoft para xerar as tricomías que van ser enviadas ó programa de pesaxe. Esta parte non se integrará no repositorio Git, xa que excede dos límites deste proxecto.

Faremos uso de ferramentas para o acceso á base de datos Paradox. - Extended Borland Database Desktop[13]: Xestión de táboas Paradox. - System.Data.OleDb[14]: Librarías de Microsoft para o acceso a bases de datos Paradox.

5.2.2 Seguridade e rendemento

Interesa considerar aspectos de seguridade e rendemento, para garantir robustez e eficiencia, como xa se fixo constar nos requirimentos non funcionais esixibles (sección 4.2). A nivel de seguridade temos que TermoPowderM será executado nunha zona específica da fábrica, o cuarto de pesadores, onde só os pesadores terán acceso ó programa. Ademais, estará conectado á LAN da empresa, que conta cunha seguridade perimetral. Ademais faremos uso do protocolo Server Message Block (SMB)[15], que integra unha protección de seu.

A nivel de rendemento, aínda que as bases de datos Paradox son extremadamente lentas, os datos a intercambiar co sistema de pesaxe non son demasiados. Preferiremos a inserción directa nas táboas Paradox por ser unha operación síncrona contra o uso de ficheiros de intercambio, que xestiona de forma asíncrona o programa TermoPowderM (4.2). Aínda así daremos opción ó programa a usa-las dúas para aumenta-la dispoñibilidade (4.2) do sistema, no caso que haxa algún problema co acceso directo ás táboas por tema de bloqueos pola interferencia do Borland Database Engine.

Capítulo 6

Implementación e probas

Neste capítulo, abordaremos a fase crítica do proceso de desenvolvemento: a implementación e as probas. Tras unha análise exhaustiva e a fase de deseño, chegamos ao punto no que transformamos as nosas ideas e especificacións en código real e funcional.

Queremos proporcionar unha visión detallada de como se levou a cabo a implementación do noso proxecto, explicando as decisións de deseño clave e como estas se traduciron no código, confirmando se así fose, as ferramentas e tecnoloxías utilizadas para levar a cabo esta etapa. Ademais, detallaremos as probas que se aplicaron para garantir a calidade e fiabilidade do software resultante.

Presentaranse exemplos de código significativos e discutiranse os desafíos encontrados durante o proceso de implementación, así como as solucións adoptadas.

6.1 Verificación do funcionamento da báscula

Tal e como se preveu que podía ocorrer, debido a que non se logrou unha comunicación coa báscula, optouse por implementar un programa de diagnose para confirmar o bo funcionamento da impresora, ve-las respostas a distintas peticións e en base a ese coñecemento xerar o ficheiro de configuración preciso para que funcionase a nosa balanza de exemplo con TermoPowderM. Co programa de diagnóstico aseguramos que a báscula funciona de forma correcta, que podemos comunicarnos con ela e que podemos avalia-las súas respostas.

6.1.1 Estrutura do código fonte

Creamos unha aplicación de consola moi simple en C#, que recibe ordes mediante a liña de comandos, para poder proba-los distintos comandos do protocolo SICS[6].

Dentro da mesma creamos unha clase `Báscula` que conterá as funcionalidades necesarias para poder diagnostica-la comunicación coa balanza.

6.1.2 Implementación de funcionalidades

Describimos a continuación as funcións que debe haber para segui-las pautas marcadas dende o deseño (capítulo 1).

- InicializarComunicacion: Inicialización da comunicación serie.
- Reset da báscula.
- Tara: Tara da báscula.
- Pesar: Pesaxe puntual.
- ListaOrdes: Inicialización, Reset e Permitir peticións por teclado.
- FinalizarComunicación: Pecha-lo porto serie.

Os nomes son autodescritivos, mais en ListaOrdes permítese a inserción de peticións á bascula por liña de comandos, o que me permite executar operacións coma a de pesaxe continua e outras como as definidas na referencia do protocolo SICS.

Vemos a continuación o exemplo da inicialización do porto serie. Fundamental para o inicio da comunicación, neste caso coa balanza. C# fai uso das librarías `System.IO.Ports` para fornece-lo acceso ós portos serie.

```
1  public bool InicializarComunicacion(string portaSerial, int
       baudRate)
2  {
3      try
4          {
5              serialPort = new SerialPort(portaSerial,
       baudRate,Parity.None,8,StopBits.One);
6              serialPort.Open();
7
8              return true;
9          }
10     catch (Exception ex)
11         {
12             Console.WriteLine("Erro ao inicializa-la comunicación:
       " + ex.Message);
13             return false;
14         }
15 }
```

Para a inserción de peticións cara a balanza fíxose uso do manual de referencia de Mettler-Toledo, que contén a listaxe dos comandos que se poden usar. Basicamente consistía en escribir sobre o porto e le-la resposta no mesmo sitio como vemos no seguinte código:

```
1     serialPort.Write("@\r\n");
2     Console.WriteLine(serialPort.ReadLine());
```

Neste caso, escribíase a orde @, que nesta báscula é a orde para inicializala.

6.1.3 Probas

Para as probas executouse unha batería de instrucións contra a balanza para obter as distintas repostas en función do estado da mesma, e desa forma poder crear o ficheiro de configuración. Entre as máis salientables están:

- @: Inicialización
- Z: Reset: Pon a zero o peso e mailas taras
- T: Tara
- S: Peso
- `SI`: Peso instantáneo
- `SIR`: Peso continuo

6.2 Creación de ficheiros de configuración

A continuación precisabamos acometer a creación do ficheiro de configuración para o uso do software herdado coa báscula en base o aprendido co programa anterior.

Este ficheiro foi creado mediante proba/erro en base ás informacións fornecidas polo programa de diagnose, e a execución contra o programa TermoPowderM que dispón dun *log* en vivo onde amosa información sobre ordes que se están a executar e a súa resposta.

A estrutura era similar en tódolos tipos de impresoras que contaban cun ficheiro de exemplo, pero cada modelo, incluso dentro de cada marca, podía variar significativamente.

Hai unha lista de instrucións, que normalmente teñen a estrutura petición/resposta, para que o programa consiga unha comunicación correcta coa impresora. Destacaremos as seguintes instrucións, as máis importantes para o correcto fluxo do traballo de pesaxe:

- `InitSend1 / InitResp1`: Petición e resposta para inicializa-la báscula. Na resposta normalmente pode aparecer o nº de serie precedido polo prefixo `I4`
- `ZeroSend / ZeroReceive`: Petición e resposta para resetea-la báscula.

- `TareSend / TaraReceive`: Petición e resposta para tara-la báscula. A resposta neste caso a de empezar por `T S` ou `T D`, que significa que vale tanto un tarado estable coma dinámico.
- `WeightMode`: Indicamos a forma de pesaxe, puntual ou continua. No noso caso interesa continua para que o programa amose por pantalla, tódalas variacións no peso.

Coa información fornecida polo programa de diagnose e mailos exemplos que existían co programa púidose xerar o ficheiro de configuración, mostrado a continuación:

```
;METTLER CONFIGURE
[General]
Description=Mettler BBK 422 - 3 XS. Max 3100 d 0.01g

[Comands]
InitSend1='@'#13#10
InitResp1='I' ;--> Vale calquera resposta que comece por I
TimeOutInit1=10000

ZeroSend='Z'#13#10
ZeroReceive='Z '
ZeroTimeOut=5000

TareSend='T'#13#10'T'#13#10
TareReceive='T S' | 'T D'  ;--> Se quixésemos ser máis restritivos,
    so "T S"
TareTimeOut=5000

WeightMode='Cont'  ;--> Indicamos que a pesaxe é continua.
WeightSend='SIR'#13#10  ;--> Instrución para a pesaxe continua.
WeightReceive='S '  ;--> Se a resposta empeza por S, a pesaxe
    acéptase. Poderíamos ser máis restritivos con "S S", que
    significa que só valerían pesadas estables da báscula segundo a
    súa configuración.
WeightTimeOut=5000
TimeOutIntPesCont=10000

WeightStartChar=5    ;--> Aquí indicamo-las posición e tamaño da
    cifra pesada
WeightNrChar=10
SignPosition=0
TareStartChar=1      ;--> Este elemento non conseguín atoparlle o
    sentido, mais aparecía en case tódolos ficheiros de exemplo e
    con outros valores, o sistema non funciona correctamente.
EndWeightSend=@     ;--> Cando queremos que deixe de pesar,
    enviamos a orde de inicialización. Este campo tiña moita
```

```
    variabilidade, pois nalgunhas impresoras había que volver enviar
    a orde de pesaxe.
29 EndWeightReceive=I
30 InitSend2=
31
32
33 [SerialComunication]  ;--> Indicámo-la configuración da
    comunicación serie.
34 Baud=9600
35 ;Values: 1200 2400 4800 9600 19200
36 Parity=NO
37 ;Values: NO MARK SPACE EVEN ODD
38 DataBit=8
39 ;Values 8 7 6 5 4
40 StopBit=1
41 ;Values: 1 2
```

O comando `Tara` foi o que máis complicacións ocasionou. Non funcionaba como debía por mor da chave, até que se fixou o valor da mesma en 1. Durante as probas no mesmo programa TermoPowderM, a primeira instrución de tarado identificábaa coma un comando erróneo, tendo que darlle dúas veces ó botón. Non se atopou a orixe do problema de por que TermoPowderM consideraba o T unha instrución errónea, pero foi solucionado no ficheiro de configuración duplicando a orde de tara: `'T'#13#10'T'#13#10`, evitando que o operario teña que tarar dúas veces.

6.3 Integración no sistema de planificación de Lavamar

A seguinte tarefa ocupouse da creación dun módulo para o uso do programa de formulación e planificación para o envío de ordes de traballo ó programa de pesaxe.

6.3.1 Estrutura do código fonte

Implementouse unha función que captura o código XML na saída dunha cadea de filtros que procesan unha Receita que é planificada un día concreto, nunha máquina concreta (adscrita ó sistema SETEX) e nunha orde concreta. Con eses datos de partido, a información vai pasando de filtro a filtro até conseguir a saída que será procesada neste caso polo sistema de Orgatex[16] de Setex.

Escoller unha Receita nunha máquina SETEX é debido é que as Receitas que van a Termoelettronica, non xeran un XML cos cálculos finais completos (cantidade de auga, cantidade de produtos químicos,...).

6.3.2 Implementación de funcionalidades

A función implementada recibe unha cadea XML, que transforma nun novo XML baseado na información de tricomías desa Receita.

Para procesar a cadea de entrada, só hai que obte-los elementos `Dyelot_Recipe`, que teñan coma atributo `N_KindOfProduct='1'`, que son os colorantes. Despois, agrupar en tricomías sempre que coincida o atributo `N_CallOff` e inserir os distintos colorantes na orde definida por `N_Counter`, obtendo por exemplo:

```
<Tricomias>
  <Tricomia idMaquinada='244113' maquina='TU02' fase='4'>
    <Dyelot_Recipe T_Dyelot='244113' N_ReDye='0'
    N_CorrectionNumber='0' N_CallOff='4' N_Counter='1'
    T_ProductName='Rojo Tubantin F2B conc.' T_ProductCode='1945'
    T_ProductShortName='0001647727' N_SpecificWeight='1'
    N_Amount='0,000989' T_Unit='kg' N_CostsPerAmount='12,85'
    N_KindOfStation='5' N_KindOfProduct='1'/>
    <Dyelot_Recipe T_Dyelot='244113' N_ReDye='0'
    N_CorrectionNumber='0' N_CallOff='4' N_Counter='2'
    T_ProductName='Pardo Tubantin ZR' T_ProductCode='2454'
    T_ProductShortName='0001668553' N_SpecificWeight='1'
    N_Amount='0,426' T_Unit='kg' N_CostsPerAmount='0'
    N_KindOfStation='5' N_KindOfProduct='1'/>
    <Dyelot_Recipe T_Dyelot='244113' N_ReDye='0'
    N_CorrectionNumber='0' N_CallOff='4' N_Counter='3'
    T_ProductName='Negro Tubantin VTF 1200' T_ProductCode='1977'
    T_ProductShortName='0001558515' N_SpecificWeight='1'
    N_Amount='0,129' T_Unit='kg' N_CostsPerAmount='0'
    N_KindOfStation='5' N_KindOfProduct='1'/>
  </Tricomia>
</Tricomias>
```

6.3.3 Probas

Seleccionáronse unhas cantas maquinadas[1], para obte-las tricomías e verificouse que os datos e maila estrutura eran correctas, porque eses elementos (`Dyelot_recipe`) son iguais no documento XML que se envía ó sistema SETEX.

[1] Unha Orde de traballo nunha máquina, data e orde concretas referenciadas por un número único `IdMaquinada`

6.4 Interface Celme-TermoPowderM

Nesta sección describiremo-la evolución da implementación da parte fundamental do proxecto, pois é o nexo de unión entre o sistema xeral da fábrica e mailo sistema de pesaxe. Describimo-la adopción das funcionalidades4.1.3 e tamén as distintas problemáticas que xurdiron.

6.4.1 Estrutura do código fonte

Como queda indicado na sección de Deseño(4), desenvolveuse unha aplicación de consola. Consta da clase principal, unha clase que levará a funcionalidade cara a base de datos Paradox da aplicación de pesaxe (clase `Celme-Paradox`) e unha clase que obterá información da base de datos corporativa (clase `Celme`), por exemplo para fornecer á base de datos Paradox de información relativa ó conxunto de colorantes activos en Lavamar.

Aínda que non é unha aplicación demasiado complexa na súa estrutura, velaquí o seu diagrama de clases:

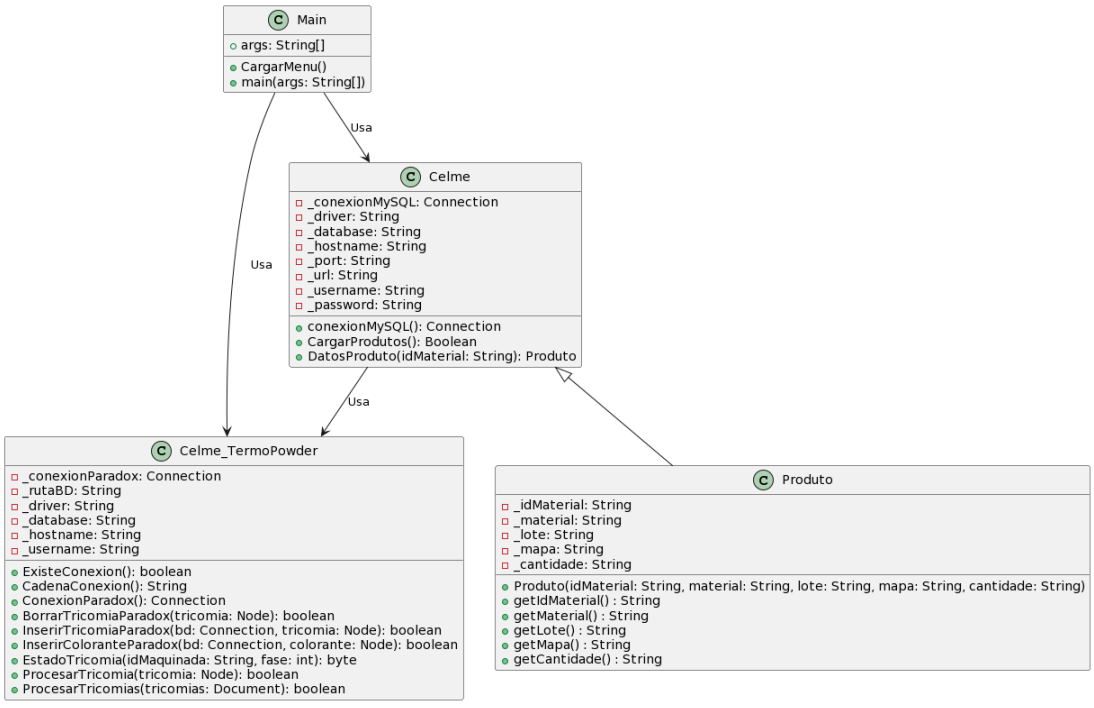


Figura 6.1: Diagrama de clases. Interface Celme-TermoPowderM

6.4.2 Implementación de funcionalidades

Tal como se describe nos requirimentos funcionais (páxina 34), a clase `Celme-Paradox` constará das seguintes funcións:

- ExisteConexion: Para ver se a base de datos Paradox (remota) está ó alcance do sistema. Úsase o protocolo SMB para a conexión.
- ConexionParadox: Obtén unha conexión activa coa base de datos.
- ProcesarTricomias: Procesa a información de entrada definida por un documento XML ou unha cadea XML.
- InserirTricomiaParadox: Insire a cabeceira da tricomía, é dicir, información como a `idMaquinada`, a fase e o número de máquina, comúns a toda a tricomía.
- InserirCoranteParadox: Insire os colorantes dunha tricomía, e as cantidades a pesar do mesmo.

Un detalle importante é que esta base de datos Paradox non ten integridade referencial, polo que o contido entre táboas é independente. Operamos coas mesmas tendo en conta este detalle.

Vemos como sería a función que conecta coa base de datos Paradox e obtén unha conexión, facendo uso das librarías `System.Data.OleDb` de Microsoft:

```
OleDbConnection ConexionParadox()
{
   if (ExisteConexion())
   {
        if (_conexionParadox == null)
        {
            _conexionParadox = new OleDbConnection
            {
                ConnectionString = CadeaConexion().ToString()
            };
        }
   }

   return _conexionParadox;
}
```

6.4.3 Problemática

Como vemos, un código moi simple, pero non exento de inconvenientes. Conseguiuse sempre conexión cara a base de datos, primeiro localmente, e despois á base de datos obxec-

tivo deste proxecto. Incluso nun primeiro intento, conseguiuse a execución dunha consulta SELECT. Pero en días posteriores, ó intentar executar unha consulta á calquera das táboas daba un erro, para o que non se atopou solución (ver imaxe 6.2): vemos que as librarías de Microsoft non podían acceder ó contido das táboas.

Os pasos que se tomaron ante esta situación foron os seguintes:

- Cambios na configuración da Borland Database Engine (BDE).
- Probas contra as táboas Paradox fora do ámbito da BDE.
- Busca doutras librarías que permitisen a conexión con Paradox en C#.

O resultado final de todas estas operacións, non foi satisfactorio, xa que non se podían executar operación (Create, Read, Update and Delete) (CRUD) polo que había que tomar unha decisión drástica: abandonar o desenvolvemento da aplicación en C#.

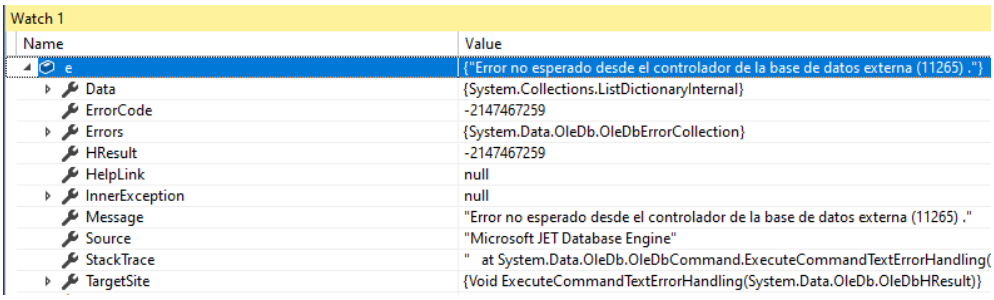


Figura 6.2: Erro provocado pola execución dun SELECT contra a BD Paradox

Analizando as opcións alternativas, decidiuse o uso de Java con linguaxe de programación despois dunha busca preliminar que confirmou a posibilidade da comunicación con bases de datos Paradox, e considerando que era unha linguaxe coñecida, polo que o tempo de adaptación sería mínimo.

Iniciamos un novo desenvolvemento, esta vez con Java, aproveitando a estrutura do código xa executado en C#, adaptándoo ó novo sistema.

6.4.4 Estrutura do código fónte

Unha vez máis, seguindo co indicado no capítulo de Deseño (capítulo 4), creamos ademais da clase principal da aplicación de consola, a clase `Celme-TermoPowder` e maila clase `Celme`, para xestionar tanto a base de datos Paradox, coma o acceso ós datos corporativos sobre colorantes.

Para o acceso ás distintas bases de datos empregáronse os seguintes conectores:

- HXTT: Paradox_JDBC40.[2]
- GoogleCode.Paradox: paradoxdriver-1.6.0.[3]
- Mysql: mysql-connector-j-8.1.0.[4]

Para o acceso á bases de datos remotas mediante o protocolo Server Message Block (SMB) usouse JCifs[17].

6.4.5 Implementación de funcionalidades

Cada elección de ferramenta ou tecnoloxía ten un impacto directo na facilidade de mantemento, no rendemento e na escalabilidade do proxecto. Así, seguindo as pautas definidas no deseño (páxina 44 e seguintes), esta vez sobre unha contorna Java, implementamo-las distintas funcións para poder cumprir cos requirimentos (páxina 34 e seguintes).

Por tanto, nesta sección, imos examinar detalladamente as opcións dispoñibles e os criterios que guían a selección das mellores solucións para alcanzar os obxectivos do proxecto. Ao facelo, estamos preparando o terreo para unha implementación exitosa, que traducirá as visións e requisitos do proxecto nun software robusto e eficaz.

- Programa Principal. Recibe ordes como argumentos, para executa-las distintas funcionalidades. Tamén recibe a información en forma de XML ou de ficheiro XML da tricomía a procesar.
- Celme-TermoPowder. Nesta clase encapsúlase a conectividade coa base de datos de TermoPowder, e as distintas operacións a realizar:
  - ExisteConexion: Para saber se temos acceso á base de datos mediante SMB.
  - ConexionParadox: Devolve unha conexión activa coa base de datos.
  - ProcesarTricomias: Procesa as tricomías dunha maquinada.
  - InserirTricomiaParadox: Dada unha conexión á base de datos e unha tricomía, insírea na táboa correspondente. De non poder, intentaría crear un ficheiro de intercambio e copialo no directorio da aplicación de pesaxe.
  - InserirCoranteParadox: Dada unha conexión á base de datos e un colorante, insíreo. De non poder, intentaría crear un ficheiro de intercambio. Como decisión tomada no momento de implementación decídese que como nas tricomías aparece o nº de lote activo, anexar ó dito lote ó nome do colorante. Isto non afecta á pesaxe

[2] http://www.hxtt.com/paradox.html
[3] https://github.com/leonhad/paradoxdriver
[4] https://dev.mysql.com/downloads/connector/j/

e o pesador, cando selecciona o colorante pode ver dito número de lote, a modo de indicación extra.

- EstadoTricomia: Devolve o estado en que se encontra a tricomía na base de datos. Os valores poden ser:
  - 0: Non rexistrada.
  - 1: En espera.
  - 2: Iniciada a pesaxe.
  - 3: Completada.
- CreaProduto: Crea un produto no almacén de colorantes dispoñibles, con información relevante (código, nome, localización...).
- BorrarProdutos: Borra os produtos do almacén de colorantes dispoñibles.
- BorrarTricomias: Borra do rexistro de tricomías tódalas tricomías pertencentes a unha maquinada.
- BorrarTricomia: Borra do rexistro de tricomías a tricomía indicada mediante o nº de maquinada e a fase.
- ListaxeTricomias: Lista as tricomías rexistradas no programa de pesaxe.

Vemos un exemplo de como consegui-la conexión coa base de datos.

```
1  private final String _driver = "com.hxtt.sql.paradox.ParadoxDriver";
2  Connection ConexionParadox()  {
3      if (ExisteConexion()) {
4          if (_conexionParadox == null ) {
5              try {
6
       java.sql.DriverManager.getConnection("jdbc:paradox:_rutaBD");
7                  Class.forName(_driver);//Aqui indicamos o nome da
       libraría para obter acceso a Paradox
8                  _conexionParadox=
       java.sql.DriverManager.getConnection(CadenaConexion());
9              } catch (Exception e) {
10                 e.printStackTrace();
11             }
12         }
13     }
14     return _conexionParadox;
15 }
```

- Celme Nesta clase encapsúlase a conectividade coas base de datos corporativa, e as funcionalidades precisas para cumprir cos requirimentos (páxina 35).

- Clase Produto: Dentro da clase nai, existe esta clase onde encapsulamos información útil do produto, nomeadamente:
  - Código do produto.
  - Nome do produto.
  - Lote do produto.
  - Mapa do produto.
  - Cantidade do vulto do produto.
- ConexionMySQL: Fornece un acceso á base de datos corporativa.
- CargarProdutos: Función que selecciona os colorantes activos na empresa e insíreos na base de datos Paradox mediante o uso da función da clase. `CelmeParadox.CreaProduto`, para ter información actualizada.
- DatosProduto: Dado un código de produto, fornece información sobre o mesmo devolvendo unha instancia da clase `Produto`.

No código seguinte, vemos como de non poder efectúa-las operacións contra a base de datos, co fin de maximiza-la dispoñibilidade do servizo, o sistema intenta crear un ficheiro de intercambio de información. Se o fluxo de execución chega a este punto significa que existe unha conexión viable mediante SMB, polo que poderemos crear un ficheiro no destino.

```
1  public boolean CrearFicheiroProduto(String idMaterial, String
       material, String zona, String coordenadas){
2      String texto="";
3      Boolean procesado=false;
4      //baseado na documentación LinkIta.doc
5      String sep="|";
6      String fin="\r\n";    //Fin de produto
7      String _idMaterial=CadeaBaleira(8);
8      String _material=CadeaBaleira(30);
9      String _zona=CadeaBaleira(2);
10     String _coordenadas=CadeaBaleira(2);
11     _idMaterial=_idMaterial.substring(0,8-idMaterial.length()) +
       idMaterial;
12     _material=material+_material.substring(0,30-material.length());
       // Aliñado á esquerda
13     _zona=_zona.substring(0,2-zona.length())+ zona;
14     _coordenadas=coordenadas.substring(0,1);
15
16     texto+= _idMaterial + sep + _material + sep + _zona + "-" +
       _coordenadas + sep + fin ;
17
18     //crear ficheiro
19     try {
```

```
20          int segundos=0;
21
22          do {
23              try {
24                  if (segundos>0) TimeUnit.SECONDS.sleep(2);
    //Retrasamos a execución 2 segundos para dar tempo a TermoPowder
    a xestionar o ficheiro prodotti
25              }
26              catch (Exception e) {
27                  e.printStackTrace();
28              }
29              segundos+=2;
30          } while ((new File(_ruta + "//prodotti.tmp").exists())||
    segundos>60) ;
31          if (segundos>60)
32              System.out.println("Tempo excedido para a creación do
    ficheiro");
33          else {
34              FileWriter fileWriter = new FileWriter(_ruta +
    "//prodotti.tmp");
35              BufferedWriter bufferedWriter = new
    BufferedWriter(fileWriter);
36              bufferedWriter.write(texto);
37              bufferedWriter.close();
38              procesado=true;
39          }
40      }
41      catch (IOException e) {
42          e.printStackTrace();
43          System.out.println("O ficheiro non se puido crear");
44      }
45      return procesado;
46  }
```

6.4.6 Probas

Para as probas empregamos o XML indicado previamente (sección 6.3.2), e seguimos o seguinte fluxo de traballo:

- Probas sobre Celme. Sobre esta clase executouse a consulta dos colorantes activos. Ditos colorantes debían ser inseridos na base de datos Paradox.
  Problemáticas:
  - Se algún colorante está repetido devolve un fallo por violación de clave (iso acontece se o produto ten 2 ou máis vultos abertos).

- Solución: Comprobar que ese colorante non existe aínda na base de datos Paradox.
- Ó face-la comprobación da existencia do colorante, o programa devolvía un erro da libraría hxtt (imaxe 6.3). Resultou que httx ten unha versión *trial* e outra de

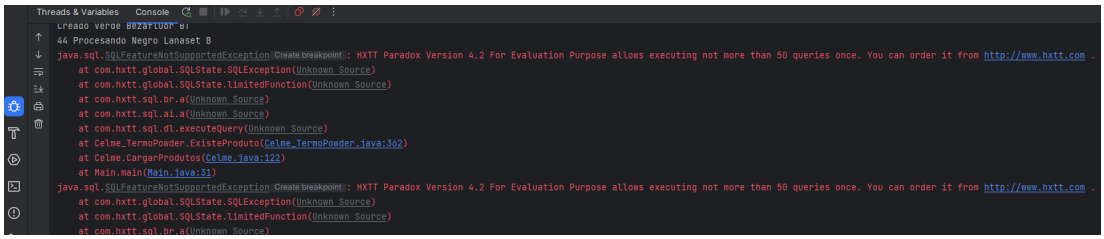


Figura 6.3: Erro por exceder as 50 consultas

pagamento. A de pagamento non está contemplada como solución, e a *trial* non permite máis de 50 consultas dentro da mesma execución.

- Primeira solución: Buscar outra libraría para conectar a Paradox. Neste caso foi `GoogleCode.Paradox`. Despois das primeiras probas con ela, funciona perfectamente ó facer operacións de consulta. O problema é que non ten implementadas as instrucións INSERT, DELETE e UPDATE de SQL. Polo tanto, tivo que desbotarse esta opción.
- Segunda solución: Modifica-lo código para evitar inserir 2 veces o mesmo colorante, xa que despois dalgunhas probas, a limitación só se refire a sentenzas SELECT. Facendo isto o programa executábase sen problema.

- Probas sobre Celme-TermoPowder
  - Execución das distintas funcións usando como dato de entrada a tricomía de exemplo (ver páxina 52).
  - Execución da función de procesamento de tricomías para un proceso con dúas tricomías.
  - Execución da función de procesamento de tricomías para unha tricomía cunha soa cor.
  - Execución da función de procesamento de tricomías para unha tricomía sen cor. Neste caso confirmouse que o sistema non insire nada no sistema remoto.

  Problemáticas
  - A maioría das problemáticas referíronse os tipos de datos na transición entre a base de datos e o motor Java para procesalos. Por exemplo un *timestamp* baleiro na base de datos é nulo na aplicación.

  - Solución: modificar código para mitigar estes erros.
  - Se a tricomía xa existía daba un erro de clave, tanto na tricomía en si (RICHDOSA) coma nos colorantes (INGDOSA).
  - Solución: No canto de volver facer un SELECT, que podía dar problemas como viamos antes, optouse por facer que antes de cada inserción se borrase calquera rastro da tricomía na base de datos remota. Con isto conseguiuse tamén evitar problemas referidos a que algún colorante estivese actualizado e non o copiase na súa nova versión. Sempre que se insire unha tricomía é a máis actualizada.

- Probas a nivel de consola Fanse probas de tódalas opcións que permite o programa a nivel de consola. Créase un artefacto `jar` para ser executado mediante Java.

  - A + Fonte de datos XML (ficheiro ou texto) -> Engadir Tricomía a traballos pendentes
  - B + IdMaquinada + Fase -> Borrar tricomía dos traballos pendentes
  - C + Fonte de datos XML (ficheiro ou texto) -> Borrar Tricomía dos traballos pendentes
  - E -> Eliminar datos relativos ó almacén de colorantes
  - L -> Listaxe de tricomias rexistradas
  - P -> Cargar produtos en almacén
  - S + IdMaquinada + Fase -> Estado da Tricomía

6.5 Conclusión

Presentamos agora unha breve recapitulación dos elementos máis destacados deste capítulo de implementación e probas.

6.5.1 Resumo dos logros na implementación

Con esta implementación, o sistema é quen de exportar cara o programa TermoPowder tricomías validadas, e que no propio programa de TermoPowder aparecen como válidas e dispoñibles para operar con elas. Tamén é quen de envia-los colorantes activos da fábrica para que estean dispoñibles no almacén de TermoPowder, ademais de fornecer de información sobre o estado das tricomías.

Cumpre, pois, cos requirimentos funcionais indicados (comentados na páxina 34 e seguintes). Tamén a implementación dun programa de diagnóstico cumpriu cos obxectivos e permitiu a creación dun ficheiro de configuración adaptado á báscula usada no proxecto (ver

capítulo 1). Por último, a modificación do módulo de planificación programa de xestión corporativo, permitiu fornecer ó interface con tricomías verificadas e ben estruturadas, cumprindo a súa función (ver capítulo 3).

6.5.2 Desafíos atopados e leccións aprendidas

A lista de desafíos foi longa, e dela destacamos as dificultades atopadas en:

- Poñer en marcha un sistema externo, que estaba nunhas pésimas condicións de conservación, é do que só se puido recupera-lo software e poñelo en marcha nun novo equipo.
- Entende-lo funcionamento do mesmo xa que nunca se vira funcionar.
- Usar unha base de datos tan antiga e con tan pouco soporte como é Paradox.
- Problemas coa interface implementada en C#, que se ben se conseguiu conecta-la coa base de datos e executar unha consulta, non volveu funcionar. Isto provocou que houbese un atraso no desenvolvemento, debido a toma de decisión de como continuar. Afectou obviamente, ó planeado no deseño, pois tivemos que optar polo uso doutras linguaxes de programación e mudaron algunhas estruturas.
- Problemáticas coas librarías de conexión, tanto en C# coma en Java.

Con tempo, interese e pensando solucións alternativas, a implementación puido continuar adiante para cumpri-las especificacións (detalladas na sección 4.1).

6.5.3 Futuras melloras

En función do observado durante o proceso de implementación as melloras futuras poderían vir por:

- Aproveita-la información xerada polo programa de pesaxe dentro do programa de planificación, integrando o estado da pesaxe nas producións, a modo informativo.
- Eliminar pesadas xa efectuadas dende o programa de planificación, para que o pesador non vexa traballos pasados sen necesidade.
- Implementación dun módulo para o uso de impresora, aínda que contraviría as boas prácticas de sustentabilidade orientadas á eliminación ou redución máxima do uso de papel.

- Implementación dunha interface co programa TermoDye (o sistema que efectúa a disolución do colorante e o envía á maquina), para que reciba as tricomías xa pesadas para que as teña dispoñibles. Deste xeito aumentaría a trazabilidade pois actualmente é un proceso sen rexistro.

Capítulo 7

Integración e validación

Abordaremos o proceso de integración dos compoñentes do sistema e as estratexias de probas empregadas para garantir o seu correcto funcionamento. O principal obxectivo desta sección é presentar de maneira detallada como se combinan as diferentes partes do proxecto nun conxunto coherente e funcional, e como se verifica a súa conformidade cos requisitos e especificacións definidas previamente.

É unha fase clave no desenvolvemento de calquera proxecto de software de calidade, e o seu contido é crucial para comprender a solidez e a fiabilidade do sistema desenvolvido no presente traballo.

7.1 Preparación da operativa de integración

Nesta etapa abordaremo-los procedementos críticos que asegurarán unha plena operatividade do programa TermoPowder. Estas tarefas iniciais son fundamentais para que o software funcione correctamente e cumpra coas súas funcións de forma precisa.

Ó seguir estes pasos de preparación, estaremos asegurando que o programa TermoPowder estea completamente listo para ser integrado na operativa de produción coa máxima eficacia e fiabilidade:

- Execución da funcionalidade de Cargar Produtos en Almacén, para que o programa TermoPowder poida operar coas tricomías que se lle envían.
- Inserción do módulo para a xeración de tricomías validadas no programa de planificación a nivel de produción.
- Inserción da execución dende ese módulo da interface Celme-Termopowder encapsulado nun `jar` para a exportación de tricomías que se foron xerando baixo demanda para que fose fornecendo ó programa TermoPowder.

- Execución dalgunhas pesadas ó azar co fin de avaliar o funcionamento correcto do programa TermoPowder con casos reais.

7.2 Probas de integración

Nesta sección, detállanse unha serie de probas deseñadas para verificar a correcta integración e funcionamento conxunto dos diferentes elementos que compoñen o sistema de pesaxe. Cada proba centra a súa atención nun aspecto específico da integración e proporciona unha validación crucial para o correcto funcionamento do sistema no seu conxunto, nomeadamente:

- Verificar que o almacén quedou nutrido co conxunto de colorantes activos en fábrica (cfg. 3).
- Verificar que despois de cada planificación dun proceso de tintura, unha tricomía é inserida na base de datos de TermoPowderM (cfg. 3).
- Verificar que despois de comezar a pesaxe dunha tricomía, ante unha consulta da interface sobre o estado da pesaxe devolve un 2 (pesaxe iniciada) (cfg. 3).
- Verificar que despois de cada pesaxe completada, a información sobre data de inicio e fin queda ben completada e ante unha consulta da interface sobre o estado da pesaxe devolve un 3 (pesaxe finalizada) (cfg. 3).

Desta maneira quedará verificada que a interacción entre os distintos módulos do sistema (Planificación + Interface + TermoPowderM) funcione correctamente e cumpre cos requirimentos, é dicir:

- Existe comunicación (cfg. 4.2).
- Seguridade ben configurada (cfg. 4.2).
- Fluxo de datos correcto (cfg. 4.2 e 4.1.1).
- Dispoñibilidade dos datos (cfg. 4.2).
- Fiabilidade dos datos (cfg. 4.2 e 4.2).
- Seguridade do sistema ó mesmo nivel que o resto da empresa (cfg. 4.2).
- Facilidade de uso. A maior parte do fluxo de traballo é transparente para o operario. O único que variará a súa forma de traballo será o pesador (cfg. 4.2).

7.2.1 Problemática

Ó comezo da execución das probas, xorde un problema común que pode ocorrer ó executar o artefacto `jar`. Ó realizar esta operación a través da liña de comandos, poden xurdir erros relacionados coa versión de Java utilizada. O erro específico é ilustrado na imaxe 7.1.

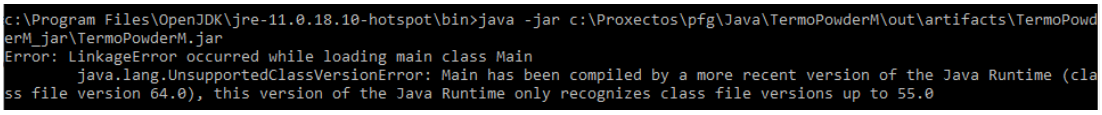


Figura 7.1: Erro pola versión de Java

A solución para solucionar este problema foi a instalación dunha versión actualizada do JDK, no noso caso a `openjdk 20.0.2 2023-07-18`.

7.3 Probas de sistema

As probas de sistema son realizadas para valida-lo sistema coma un todo e garantir que atenda ós requirimentos funcionais e non funcionais.

O foco é que o sistema funcione conforme ó esperado no seu ambiente de implantación.

7.3.1 Probas de estrés

Aínda que o sistema recibe as tricomías baixo demanda e de forma secuencia á saída do pipeline(5.4, forzamos a inserción de 12 tricomías, unha tras outra. Isto provocaba un erro en Java que se subsanou retrasando medio segundo o envío da seguinte tricomía.

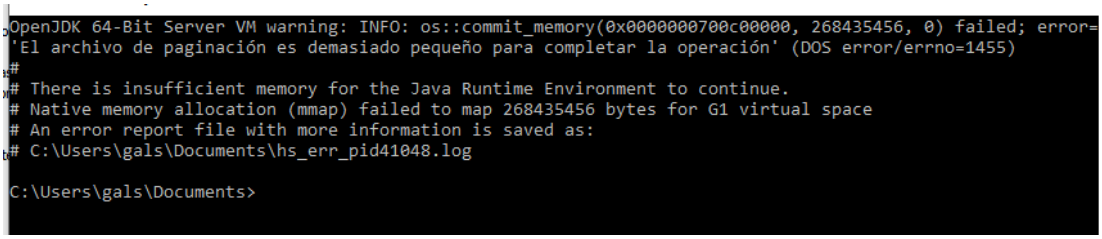


Figura 7.2: Erro de paxinación Java

Executada a proba entón, devolveu estes valores:

```
Hora inicial da proba: 07/09/2023 15:03:56.
Hora final final da proba: 07/09/2023 15:04:02.
```

A táboa 7.1 presenta os tempos de execución para as insercións mencionadas.

Vemos que se cumpre sobradamente o requirimento de rendemento (cfg. 4.2), pois a inserción dunha tricomía é de medio segundo aproximadamente. Cunha das tricomías medimo-lo

Iteración	Tempo
0	
1	15:03:56
2	15:03:56,545
3	15:03:57,091
4	15:03:57,636
5	15:03:58,182
6	15:03:58,727
7	15:03:59,273
8	15:03:59,818
9	15:04:00,364
10	15:04:00,909
11	15:04:01,455
12	15:04:02

Táboa 7.1: Tempos de inserción completa de 12 trícomías na BD Paradox

tempo de creación do ficheiro de intercambio en destino, é devolveu un tempo de 0,431 segundos, cumprindo así coa condición de rendemento estipulada tamén (cfg. páxina 35).

7.3.2 Probas de aceptación dos usuarios

Forzouse unha proba completa coa tricomía empregada durante as probas (cfg. 4.1.1), onde se planificou un proceso. Ó entrar no programa de pesaxe, figuraba dita tricomía e procedeuse a pesala iterando polo ciclo completo de pesaxe, sen incidencias. O peso rexistrado era correcto e foi validado polo pesador (cfg. 4.2).

O sistema de pesada púxose en castelán, xa que estaba configurado para aparecer en inglés, polo que se verificou o soporte multilingüe (cfg. 4.2).

7.3.3 Probas de seguridade

Para verificar a seguridade do acceso á base de datos (cfg. 4.2), inseriuse o nome de usuario/chave distinto do admitido para ó acceso por SMB para verificar que a autenticación estaba ben configurada, non obtendo acceso remoto (cfg. 4.2).

7.4 Conclusión

A través das distintas probas de integración e sistema conseguimos verifica-lo funcionamento correcto do sistema coma un todo, validando o cumprimento dos requirimentos(4) esixidos, e maila aceptación do usuario.

Deste xeito, gañamos a confianza necesaria para que o sistema quede listo para a súa instalación na localización prevista e posta en marcha en produción.

Capítulo 8

Despregamento, posta en marcha e mantemento

Esta sección é crucial para que o proceso desenvolvido estea operativo na contorna de produción, xa que buscaremos garantir que tódolos elementos do sistema funcionen correctamente e que os usuarios poidan comezar a utilizalo de forma efectiva. Consideraremos a interacción co entorno preexistente e aseguraremos que o sistema se integre de forma adecuada coa infraestrutura e as ferramentas existentes. Proporcionaremos unha guía completa para asegurar que o software sexa despregado con éxito e que se manteña de maneira efectiva ao longo do seu ciclo de vida operativo. Unha vez que o sistema está en funcionamento, é vital manter unha vixilancia continua para resolver calquera problema que poida xurdir así como abordar cuestións de actualización e mellora do software a medida que evoluciona.

8.1 Despregamento

Esta sección aborda a fase crítica do despregamento do sistema en produción. Nel proporciónase unha visión completa de como se estableceu a infraestrutura para a posta en marcha e operación do sistema de pesaxe, detallando os aspectos técnicos e prácticos necesarios para o seu correcto funcionamento dentro do entorno produtivo da empresa.

Neste apartado tamén se describirán en detalle tanto o hardware como o software utilizados no sistema de pesaxe. O equipo que executa o programa TermoPowder preséntase coas súas características chave, desde o procesador ata os periféricos e sistemas operativos específicos. Ademais, detállase a instalación de software adicional, como o Borland Database Engine (BDE) e o eXtend Borlad Database Desktop (XBDD), necesarios para executar o programa de pesaxe e xestionar as bases de datos Paradox[18] de maneira eficiente.

8.1.1 Preliminares

Habilítase unha zona específica na mesa de pesadores para aloxar o ordenador e os periféricos esenciais, permitindo así un uso cómodo para os pesadores. Fornécese ó ordenador dunha conexión á rede LAN da empresa.

8.1.2 Configuración da rede

Un dos aspectos fundamentais durante esta etapa é a configuración da rede, que permite a comunicación efectiva entre o sistema de pesaxe e outros compoñentes do entorno de producción. Para iso, asígnanse parámetros específicos, como o nome do ordenador, o enderezo IP, o grupo de traballo e a compartición SMB[15], que facilitan a conectividade e a transferencia de datos entre os diferentes sistemas incluído o acceso á rede LAN da empresa para garantir unha comunicación fluída e segura.

A rede foi configurada cos seguintes parámetros:

- Nome do pc: ”Pesaxe”
- IP: 192.168.0.87/24
- Grupo de traballo: TERMO
- Compartición Samba: //192.168.0.87/Termoelettronica
- Usuario: termo
- Chave: ––

8.1.3 Hardware e software

O equipo sobre o que corre o programa TermoPowder consta das seguintes características:

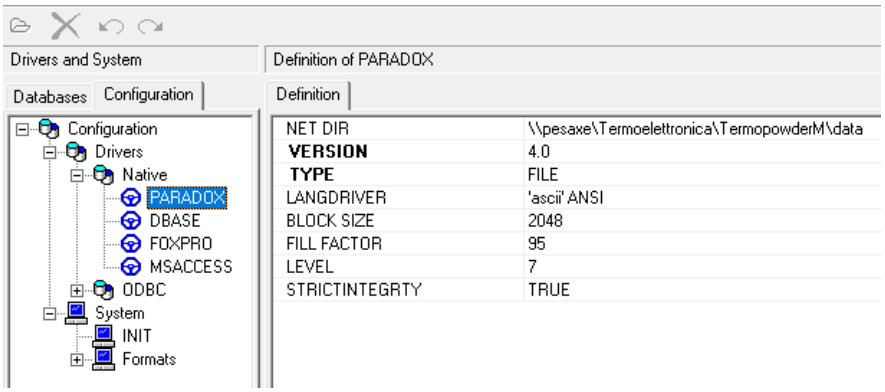
- Procesador Intel Celeron E1500 2,20GHz
- Memoria 4GB
- Sistema Operativo Windows 10 Pro for Workstations v.1919 64 bits
- Pantalla táctil
- Lector de código de barras
- Conectores de porto serie rs-232

Ademais engadiuse o seguinte software para poder executa-lo programa de pesaxe e algunha utilidade para le-las bases de datos Paradox:

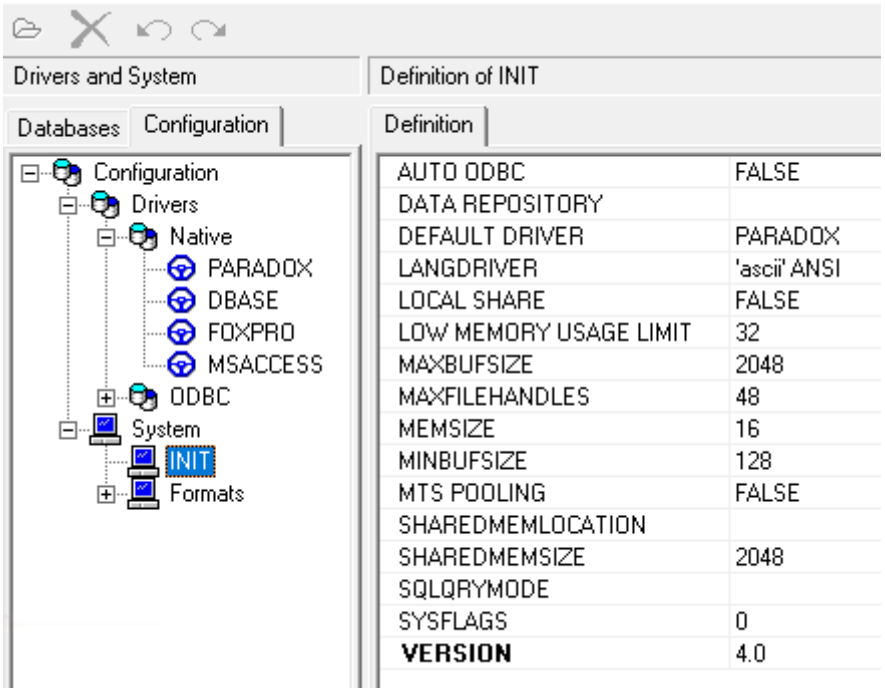
- Borland Database Engine (BDE) x32 bits v.5.01.

8.1.4 Configuración de BDE

Para que a base de datos Paradox do sistema de pesaxe funcione correctamente hai que configurar unha serie de parámetros dentro do Borland Database Engine, para que xestione correctamente unha base de datos Paradox en rede.



(a) Configuración Drivers-Native-Paradox



(b) Configuración System-Init

Se non se configura correctamente o BDE, o programa de pesaxe non pode arrincar.

- eXtend Borlad Database Desktop (XBDD), que é unha versión mellorada do Borland Database Desktop (BDD). Trátase dunha ferramenta para acceder e xestionar táboas Paradox: índices, compactación, creación e modificación de táboas, etc.

8.1.5 Incidencias

Non houbo demasiadas incidencias nesta parte do proxecto. O principal problema foi que houbo que habilitar Server Message Block (SMB) v1.0[1] no ordenador de pesaxe, para poder conectar o programa de dosificación durante a reconstrución do sistema (ver sección 5.1.1) para proba-la comunicación entrámbolos dous equipos, xa que este corre sobre Windows XP SP3.

No futuro a intención é volver deshabilitalo xa que o obxectivo é que o programa de pesaxe non teña comunicación co sistema de dosificación, e o resto de ordenadores corren na súa maioría sobre Windows 10.

8.1.6 Contorna de produción

Para proporcionar unha visión completa, inclúense fotografías da contorna de produción na que se atopa o sistema de pesaxe no cuarto de pesadores. Estas imaxes (8.2 e 8.3) ilustran a disposición física e a ubicación do equipamento dentro dese entorno, o que facilita a comprensión da infraestrutura implementada.

Figura 8.2: Cuarto de pesadores.

[1] https://learn.microsoft.com/es-es/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3?tabs=server

Figura 8.3: Cuarto de pesadores (detalle)

8.2 Mantemento

Esta sección detalla as medidas e estratexias de mantemento necesarias para garantir o funcionamento eficaz e fiable do sistema de pesaxe, así como propostas de ideas de mellora para continuar optimizando a súa operativa a longo prazo. Para isto, estableceranse medidas concretas que abarcan desde o rexistro de incidencias ata a implementación de melloras e actualizacións.

8.2.1 Previsión de actuacións

A lista de actuacións que habería que executar sería:

- *Rexistro de incidencias.* Para manter un sistema eficiente, é esencial contar cun sistema de rexistro de incidencias. Deste xeito, é posible documentar e analizar os problemas que poidan xurdir no proceso de pesaxe, nomeadamente:
  - Non aparecen as tricomías para seren pesadas.
  - Os datos da tricomía non concordan.
  - A base de datos Paradox non funciona correctamente por mor de índices corruptos.
  - O programa non se executa seguindo a secuencia referida nos requisitos funcionais (ver páxina 33).

  - Outras incidencias percibidas polos pesadores.

- *Corrección de erros despois do estudo das incidencias.* Unha vez rexistradas as incidencias, procederase á súa corrección. Isto require un estudo detallado de cada caso para identificar as solucións axeitadas. As correccións permitirán manter unha operación fluída e eficaz do sistema de pesaxe. Para garantir unha medición precisa e fiable, tamén se establecerá un calendario de calibración das básculas conectadas. Deste xeito, aseguraremos que os instrumentos de pesaxe funcionen dentro dos estándares de precisión requeridos.

- *Etiquetaxe correcta das caixas activas de colorante.* Opcionalmente, aínda que moi recomendable, débese prestar especial atención á etiquetaxe correcta das caixas activas de colorante. Esta medida non só garante unha trazabilidade adecuada, xa que asegura a inserción de datos fiables no programa de pesaxe.

- *Actualización da documentación.* Será esencial manter actualizada a documentación relacionada co sistema de pesaxe. Isto inclúe manuais de usuario, guías de configuración e toda a información relevante para o correcto funcionamento e mantemento do sistema.

8.2.2 Melloras futuras

Ademais das medidas de mantemento, é importante considerar posibles melloras futuras. A mellora continua do software é esencial para asegurar a súa eficacia a longo prazo. Neste sentido, é fundamental manter unha canle de comunicación aberta co persoal que utiliza a aplicación, co fin de recoller información sobre posibles melloras que se poderían implementar no futuro. Ademais, a posibilidade de integrar o sistema de pesaxe co resto de básculas dispoñibles para a produción é unha opción a ter en conta, xa que maximizaría o rendemento e a utilidade desta aplicación no conxunto da contorna de produción.

En vista da mellora continua do software, é interesante o intercambio de información co persoal que vaia usar este aplicativo co fin de recoller información sobre melloras que se poderían implementar no futuro.

Capítulo 9

Conclusións

O proxecto chegou a súa fin de forma exitosa, xa que se cumpriron o conxunto de requirimentos de usuario, acabouse sen sobrepasa-la data límite e sen gastos extras non contemplados. Ademais, a realización deste proxecto achega unhas cantas leccións aprendidas moi orientadas á reenxeñaría, que nunha empresa onde o escenario produtivo vai evolucionando tanto con tempo, é fundamental para o reaproveitamento de sistemas e a súa interconexión cun espectro moi diverso de tecnoloxías. Completamos este derradeiro capítulo desta memoria con algunhas reflexións personais.

9.0.1 Obxectivos atinxidos

Este traballo desenvolveuse no seo dunha pequena empresa, Lavamar, que leva a cabo a súa actividade no eido do sector téxtil, en concreto dedicada aos procesos de lavado e tintura de prenda confeccionada. Partiamos dun escenario onde non había ningún tipo de rexistro de pesaxe dos colorantes empregados para a tintura, co que iso repercute na trazabilidade do sistema produtivo, e onde había un sistema de guiado de pesaxe en malas condicións nun almacén dende fai moitos anos. Ese sistema adquirírase para rexistrar pesadas pero nunca chegara a empregarse, polo que se fose posible integralo dentro do fluxo de traballo solucionaríase o problema de trazabilidade, alén de amortizar un investimento xa case esquecido.

A situación despois do traballo realizado é que temos un sistema de pesaxe operativo, que rexistra as pesadas realizadas dun xeito fiable e verificable, que guía na pesaxe ó pesador e que fornece coa súa operativa a trazabilidade no fluxo produtivo. O sistema de pesaxe queda entregado para o seu uso en produción en Lavamar, e seguirá sendo supervisado, xa dende un ámbito de mantemento, tal é como se xustifica no ciclo de vida escollido.

9.0.2 Leccións aprendidas

No noso caso, entre as leccións aprendidas de maior impacto destacan a interacción con bases de datos moi antigas, sempre complicadas de xestionar pola súa propia implementación

e pola falta de documentación, e o uso dun hardware como as básculas, cuns protocolos de comunicación cos que nunca se traballara, e que grazas a este proxecto abren un novo marco de actuacións cara o futuro para novos proxectos, todos orientados a controla-lo sistema produtivo, base fundamental do traballo ben feito.

9.0.3 Futuras liñas de traballo

Como traballos futuros moi relacionados con este (alén dos xa mencionados nos capítulos 6 –sección 6.5.3– e 8 –sección 8.2.2), e grazas o coñecemento adquirido, está a creación de novos ficheiros de configuración para poder usar outras impresoras que están en uso na empresa. Tamén é viable a posibilidade de conectar o sistema de disolución do colorante, que recibiría a lista de tricomías xa pesadas. O determinante nestes pasos do ciclo de produción sempre vai se-lo mesmo: a trazabilidade e rexistro de calquera operación. Se hai un problema e hai rexistro do mesmo, pois seguramente haxa unha solución para que non se repita.

9.0.4 Reflexións personais

Nun mundo onde se tende a facer cousas de usar e tirar, non está de máis mirar ó noso redor e ver que hai cousas que poden ter unha segunda vida que teña un impacto significativo no fluxo de traballo. Os nosos clientes cada vez buscan maior trazabilidade no noso traballo, polo que o rexistro de cada paso en produción faise imprescindible. Igual agora este programa non era imprescindible, mais nun futuro próximo íao ser, e igual daquela habería que pagar por el se non se tivese realizado, coa antelación e tempo suficiente, un proxecto coma este. Alén diso, tal como funcionan as aplicacións e sistemas empresariais, moi probablemente habería que pagar un plus por cada módulo ou funcionalidade que se quixese engadir. Malia isto, posiblemente o impacto económico oculto deste traballo non teña un recoñecemento na remuneración actual nin futura de quen o levou a cabo, mais permanece o orgullo profesional e ter aprendido multitude de cousas útiles para o futuro.

Aprendín moito neste curso. Non precisaba facelo, pois xa era enxeñeiro en sistemas e tiña traballo. Pero considerei poñerme a proba, ver o meu nivel, e apunteime o CAIT. Valoro moi positivamente as materias que cursei, especialmente Arquitectura do Software e Xestión de proxectos. Gústame aprender de quen sabe máis ca min e non é difícil aprecia-lo nivel da docencia. Como dixen, eu son enxeñeiro de sistemas e fixen moito traballo de enxeñeiro de sistemas, pero acabei desenvolvendo software con múltiples finalidades, e por iso mesmo me decidín en segui-la mención en Enxeñaría do Software, porque ó fin é cabo é no que basicamente desenvolvín o meu traballo na empresa na que levo dende 1999. Tamén, porque case sempre tiven que traballar nun equipo composto por unha persoa. Iso significa que aprendes, pero desenvolves moitos "vicios" tamén. Gustoume ver que moitas solucións arquitectónicas que eu deducín pola experiencia, foron validadas neste curso. Gustoume aprender que aínda

as podo refinar máis. Gustoume aprender a xestionar o desenvolvemento dende os alicerces, porque aínda que xa o fixese, desde logo non era dunha maneira formal.

Creo que hai moita xente que sairá desta facultade e non acabará nunha grande empresa, cun amplo equipo de traballo, altamente especializado. Moitos acabarán nunha Peme, e se cadra terán que traballar sós. E poden facer un traballo fantástico que permita, na porcentaxe que lles toque, que a empresa na que traballen poida sobrevivir neste ámbito onde só aguantan as máis fortes. Pero non é doado. Penso que se podería dedicar algo de tempo de docencia a como poder orientar a eses alumnos potenciais traballadores de Pemes a traballar nun entorno laboral que no fondo non vai comprender a dificultade do teu traballo, que seguramente o minorizarán, e no que un proxecto non se fai nun día e que hai que xestionalo. Eu xa aprendín a base de anos, pero se un xa vai aprendido, igual ten menos decepcións.

Ó final, no fondo, este proxecto reflicte o que debe facer un enxeñeiro hoxe en día: unha reenxeñaría de si mesmo cada día.

Apéndices

Apéndice A

Orzamento dun sistema de pesaxe

EXEMPLO de orzamento para un sistema de pesaxe no mercado

Oferta 217_1/2020 6 Agosto 2020

Pesado Guiado con enlace a sistema de control Orgatex

Instalación del software de pesado guiado 3GI en un ordenador preparado por el cliente.

Adecuación de las básculas proporcionadas por el cliente (Mettler Toledo BBK 422 - 3 XS, Gibertini EU-C 4002 y la báscula grande SV-16K es de 16K) para enlace con el programa 3GI

Enlace del programa de pesaje guiado con el sistema de gestión de máquinas de tinte Orgatex.

Se utilizaran los controladores Setex 575 para enlazar el pesado guiado a las máquinas de tintura.

Sw 3GI de pesado guiado	**Euro 1.900,00**
3 Controladores Setex 575 con modulo FM32A	**Euro 6.000,00**
Gestión de pruebas y puesta en producción	**Euro 1.500,00**
Desarrollo sw para conexión de básculas existentes al programa	**Euro 2.600,00**
Enlace con Orgatex	**Euro 2.800,00**

Figura A.1: Detalle do orzamento para un sistema de pesaxe

Apéndice B

Intercambio de datos por rede

TermoPowderMan: network data exchange.

The controlling PC of TermoPowderMan can receive the products configuration, the recipes batches, and can send back the done products batches. To enable the data exchange the extra parameter "/r" must be added at the calling parameters. By example:

C:\WinMag\Runmag.BAT xxx /r.

Where xxx identificate the local language (fra, esp, ita, eng, …). Usually the program activation is made with a link on the MSWindows desktop. The extra parameter must be added in the properties of the link.

Specifications

The data exchange is made via a local area network (LAN) and ascii files, readable with a normal text editor, like notepad.
The exchange folder must be set in the [Rete] section of the configuration file C:\WinMag\Termo.ini, setting the "PathRete" key, by example:

[Rete]
PathRete=F:\TERMO\DATI

In the field specifications below the letter 'C' indicates an alfabetical field, A indicates un alfanumerical field, N indicates a numeric filed.

Recipes Batches file

The exchange file name must be set in the [Rete] section of the configuration file C:\WinMag\Termo.ini, setting the "PathDosaggi" key, by example:

[Rete]
PathRete=F:\TERMO\DATI
PathDosaggi=magazz.tmp

The file can contains more batches, one for each row, with each field separated by the separator " | " (pipe). Hereafter is the format:

Action	1 byte ascii + 1 byte separator C=delete stored batch A=store the batch	(C)
Batch Order Number	12 byte ascii + 1 byte separator	(N)
Machine Number	2 byte ascii (1-99) (right aligned) + 1 byte sep.	(N)
Phase/Step Number	3 byte ascii (1-999) (right aligned) + 1 byte sep.	(N)
for 20 products:		
Product Code	8 byte ascii (right aligned) + 1 byte sep.	(N)
Product Quantity	9 byte ascii, fomat "6.2", right aligned,+ 1 byte sep.	(N)

Actally the separator is skipped, so any ascii character can be used.

An example :

```
A|16           |11|001|00000011|    500.00|00000013|    300.02|
```

is a batch for the order 16, step 1 , machine 5 , 300.00 g of product 00000011 and 100.00 g of product 00000013.

Done products batches

The exchange file name must be set in the [Rete] section of the configuration file C:\WinMag\Termo.ini, setting the “PathPesate” key, by example:

[Rete]
PathRete=F:\TERMO\DATI
PathDosaggi=magazz.tmp
PathPesate=pesate.txt

The file has the below format:

Batch Order Number	12 byte ascii (left aligned) + 1 byte sep.	(N)
Machine Number	2 byte ascii (1-99) (righth aligned) + 1 byte sep.	(N)
Product Code	8 byte ascii (allineato a destra) + 1 byte sep.	(N)
Product target quantity	9 byte ascii, fomat “6.2” (righth aligned) + 1 byte sep.	(N)
Product actual quantity	9 byte ascii, fomat “6.2” (righth aligned) + 1 byte sep.	(N)
Day	2 byte ascii (1-99) (righth aligned) + 1 byte sep.	(N)
Month	2 byte ascii (1-99) (righth aligned) + 1 byte sep.	(N)
Year	4 byte ascii (1-9999) (righth aligned) + 1 byte sep.	(N)
Hour	2 byte ascii (1-99) (righth aligned) + 1 byte sep.	(N)
Minutes	2 byte ascii (1-99) (righth aligned) + 1 byte sep.	(N)

An example:

```
12          |01|     123|    10.00|     9.77|03|08|2000|14|45|
```

Each line is terminated by CR+LF (ascii 13+ascii 10). The row length is then fixed to 61 bytes.

Apéndice C

Estrutura das táboas da base de datos Paradox

Neste anexo vemo-la estrutura das táboas coas que traballará a aplicación para fornecer cos datos das tricomías ó programa de pesaxe

Structure of table:
C:\Termoelettronica\TermopowderM\data\RICHDOSA.DB

Table info: .....

No	Field Name	Type	Size	Required	Key
1	COMM	A	12		✓
2	NMAC	S			✓
3	FASE	S			✓
4	INIZ	@			
5	FINE	@			
6	VALIDA	L			
7	BARCODE	A	20		
8	RICE	A	9		
9	KG	N			

No	Index Name	Index Fields
1	DataInizio	INIZ

Structure of table:
C:\Termoelettronica\TermopowderM\data\INGDOSA.DB

Table info: .....

No	Field Name	Type	Size	Required	Key
1	COMM	A	12		✓
2	NMAC	S			✓
3	FASE	S			✓
4	QUANT	N			✓
5	NUMP	N			✓
6	CODICE	A	8		
7	DESC	A	30		

No	Index Name	Index Fields

Structure of table:
C:\Termoelettronica\TermopowderM\data\STOCK.DB

Table info: .....

No	Field Name	Type	Size	Required	Key
1	PROD	A	8		✓
2	DESP	A	30		
3	VLVL	A	20		
4	DCAR	D			
5	QCAR	N			
6	GMIN	N			
7	GATT	N			
8	DIFFER	N			
9	GATTMP	N			
10	CONCEN	N			
11	BARCODE	A	30		
12	CAPS	N			

No	Index Name	Index Fields
1	byPROD	PROD

Relación de Acrónimos

BDD Borland Database Desktop. 71

BDE Borland Database Engine. 38, 46, 55, 69–71

CAIT Adaptación al Grado para Ingenieros Técnicos. 76

CRUD (Create, Read, Update and Delete). 55

LAN Local Area Network. 36, 38, 46, 70

Peme Pequena e mediana empresa. 77

SMB Server Message Block. 27, 29, 46, 54, 56, 58, 67, 70, 72

XBDD eXtend Borlad Database Desktop. 69, 71

Glosario

COM Communication Port.. 43

fase Indica un lugar concreto dentro da execución da receita onde se engaden os produtos (auxiliares ou colorantes) dentro da máquina.. 20, 54, 57

Git Git é un sistema de control de versións (VCS) que traza os cambios en ficheiros e permite a coordinación de traballo en grupos de moitas persoas.. 46

idMaquinada Nº único para definir unha orde de traballo, nunha máquina concreta nun tempo concreto.. 27, 54

Java Linguaxe de programación Java.. v, vi, 55, 56, 60–62, 66

JDK Java Development Kit.. 66

maquinada Etapa da orde de traballo definida pola idMaquinada.. 52, 56, 57

Orde de traballo Conxunto de operativas a executar, incluído o proceso descrito pola receita, sobre un lote de roupa.. 52

Paradox Paradox é unha base de datos relacional.. v, vii, 26, 28, 35, 38, 44, 46, 53–55, 58–60, 62, 67, 69–71, 73

Receita Código fonte onde vén descrito o proceso físico-químico que hai que levar a cabo para obter unha tintura correcta de roupa.. 2, 3, 34, 44, 51, 52

tricomía Conxunto de colorantes que se pesan e engaden xuntos.. v, 19–21, 26–29, 34, 35, 37, 40, 43–46, 52, 54, 56, 57, 60–67, 73, 76

XML Linguaxe de etiquetaxe estendíbel (en inglés, Extensible Markup Language.. 8, 34, 35, 44, 46, 51, 52, 54, 56, 59, 61

Bibliografía

[1] Wikipedia, "V-model (software development)," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://en.wikipedia.org/wiki/V-model_(software_development)

[2] B. O. del Estado, "Resolución de 5 de octubre de 2011, de la universidad de a coruña, por la que se publica el plan de estudios de graduado en ingeniería informática," 11 2011. [En liña]. Dispoñible en: http://www.boe.es/boe/dias/2011/11/18/pdfs/BOE-A-2011-18126.pdf

[3] Inditex, "The list by inditex," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://app.powerbi.com/view?r=eyJrIjoiNDI5ZGY2MzQtYTdiMS00MDk1LTk1NTYtZDNiMDc4MmYwNmQ4IiwidCI6ImM4ZThiZGI2LT

[4] ——, "Zero discharge," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.roadmaptozero.com/

[5] S. Brown, "The c4 model for visualising software architecture," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://c4model.com

[6] Mettler-Toledo, "Mt-sics interface commands," 2019, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.mt.com/dam/P5/labtec/17_Miscellaneous/RM_Advanced_and_Standard_Level_Balances_SICS_EN.pdf

[7] M. Toledo, "Balanzas compactas bba422 / bba425 / bbk422," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.mt.com/mt_ext_files/Editorial/Generic/8/BA_BBK422_35paint_Editorial-Generic_1134123313362_files/22011383B.pdf

[8] Embarcadero, "Bde overview," 2022, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://docwiki.embarcadero.com/RADStudio/Alexandria/en/BDE_Overview

[9] Termoelettronica, "Termoelettronica," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.termoelettronica.it

[10] S. S. textile computer Gmb, "Setex," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.setex-germany.com/en/

[11] Microsoft, "Visual studio," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://visualstudio.microsoft.com/es/

[12] Wikipedia, "Visual basic for applications," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://en.wikipedia.org/wiki/Visual_Basic_for_Applications

[13] Proyectoa.com, "Borland database desktop. descarga," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://proyectoa.com/download/borland-database-desktop-7-0/

[14] Microsoft, "System.data.oledb espacio de nombres," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://learn.microsoft.com/es-es/dotnet/api/system.data.oledb

[15] Wikipedia, "Server message block," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://en.wikipedia.org/wiki/Server_Message_Block

[16] S. S. textile computer Gmb, "Mes-orgatex," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.setex-germany.com/en/products/mes-orgatex

[17] T. J. Project, "The java cifs client library," consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://www.jcifs.org/

[18] Wikipedia, "Paradox (database)," 2023, consultado o 11 de setembro de 2023. [En liña]. Dispoñible en: https://en.wikipedia.org/wiki/Paradox_(database)