

PRESENTACIÓN SQUAD 2

CASO DE NEGOCIO ALKEMY - TDF

PRINCIPALES TECNOLOGÍAS USADAS EN EL PROYECTO



INTRO

- **Introducción a Python, Odoo y Desarrollo con Docker y PostgreSQL**
- *Bienvenidos, hoy profundizaremos en el fascinante mundo del desarrollo empresarial, explorando Python, Odoo y cómo, mediante Docker y PostgreSQL, construimos soluciones eficientes y escalables.*

Python:

- *Python, conocido por su elegancia y simplicidad, es un lenguaje de programación de propósito general.*
- *Su amplio ecosistema de bibliotecas facilita el desarrollo en diversas áreas, desde aplicaciones web hasta análisis de datos.*
- *En el desarrollo de Odoo, Python se destaca como el lenguaje principal, permitiendo la creación de módulos y extensiones de manera efectiva.*

Odoo:

- *Odoo, un ERP de código abierto, abarca múltiples áreas empresariales: desde ventas y compras hasta recursos humanos y contabilidad.*
- *Su diseño modular permite la adaptación precisa a los requerimientos de cada empresa.*
- *Odoo emplea una arquitectura MVC (Modelo-Vista-Controlador), lo que simplifica el desarrollo y la personalización*

Docker y PostgreSQL:

- *Docker simplifica la gestión de entornos de desarrollo y producción mediante contenedores, garantizando la consistencia y la portabilidad de la aplicación.*
- *PostgreSQL, elegido como la base de datos por defecto en Odoo, ofrece robustez y rendimiento.*
- *La combinación de Docker y PostgreSQL facilita el despliegue escalable de aplicaciones Odoo en diversos entornos.*

Desarrollo con Python en Odoo:

- *Python actúa como el lenguaje de implementación en Odoo, proporcionando una sintaxis limpia y estructuras de código claras.*
- *El desarrollo de módulos personalizados y la expansión de funcionalidades existentes son tareas intuitivas gracias a Python.*
- *Odoo utiliza ORM (Mapeo Objeto-Relacional) para gestionar la interacción con la base de datos, simplificando las operaciones de CRUD (Crear, Leer, Actualizar, Eliminar).*

En resumen, Python y Odoo forman una alianza poderosa en el desarrollo empresarial. A través de Docker y PostgreSQL, creamos un entorno eficiente y escalable. Hoy, exploraremos cómo estas tecnologías colaboran para construir soluciones empresariales sólidas. ¡Comencemos nuestro viaje!

environment:

- PGHOST=odoo15-db:5438
- ODOO_CONFIG_DIR=/etc/odoo/
- ODOO_CONFIG=odoo.conf
- CRYPTOGRAPHY_DONT_BUILD_RUST=1

Para actualizar los modulos de la base de datos

Enterprise

command: -c /etc/odoo/odoo.conf --log-level='debug_rpc_answer'

volumes:

odoo15-web-web-data:

odoo15-web-db-data:

```
services:

odoo15-db:
    #restart: always
    image: postgres:10
    container_name: odoo15-db
    environment:
        - POSTGRES_PASSWORD=odoo15
        - POSTGRES_USER=odoo15
        - POSTGRES_DB=postgres
        - PGDATA=/var/lib/postgresql/data/pgdata
    ports:
        - "5438:5432"
    expose:
        - "5438"
    volumes:
        - odoo15-web-db-data:/var/lib/postgresql/data/
        - "/etc/timezone:/etc/timezone:ro"
        - "/etc/localtime:/etc/localtime:ro"
```

```
odoo15-web:
  #restart: always
  container_name: odoo15-web
  build:
    context: odoo-15.0
  links:
    - odoo15-db
  depends_on:
    - odoo15-db
  ports:
    - "9005:9005"
    - "7989:7989"
  expose:
    - "9005"
    - "7989"
  volumes:
    - odoo15-web-web-data:/var/lib/odoo
    - "/etc/timezone:/etc/timezone:ro"
    - "/etc/localtime:/etc/localtime:ro"
    # Modificar segun la ubicacion de los modulos
    - ./addons:/mnt/extra-addons

  stdin_open: true
  tty: true
```

```
h2s@h2s-VirtualBox:~$ docker pull odoo:14
14: Pulling from library/odoo
Digest: sha256:0123fb5ec226a75b4c6f3166c6ebee8c133adea50efb49378b1e5b6a52dab16f
Status: Downloaded newer image for odoo:14
docker.io/library/odoo:14
h2s@h2s-VirtualBox:~$ docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
bd897bb914af: Pull complete
7f145551e8b9: Pull complete
d21bf1caa4a5: Pull complete
7d593d17cf79: Pull complete
c468fd1ea184: Pull complete
cd96a2d4842d: Pull complete
12fbff9d6306: Pull complete
59e3d6202528: Pull complete
596c65fd344f: Pull complete
9ad62c5f06a9: Pull complete
bf659d84c20b: Pull complete
42945cf6db15: Pull complete
13f7bf05d84e: Pull complete
Digest: sha256:175ff61a978bc829954062714ec673912b5bfbbd17f012a0f474945200cd48fb
```

HERRAMIENTAS FUNDAMENTALES PARA EL DESARROLLO



- Visual Studio Code es un editor de código potente y versátil que proporciona un entorno de desarrollo eficiente y amigable para programadores de diferentes niveles de experiencia y que trabajan en diversos proyectos.
- GitHub es una plataforma integral que facilita la colaboración y el desarrollo de software, proporcionando herramientas para el control de versiones, seguimiento de problemas, colaboración efectiva y muchas otras funcionalidades esenciales para los desarrolladores y equipos de desarrollo.



NQNseba / squad2_odoo_gradein



Type ⌘ to search



Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

main

squad2_odoo_gradein / gradein_services /

Go to file



Add file



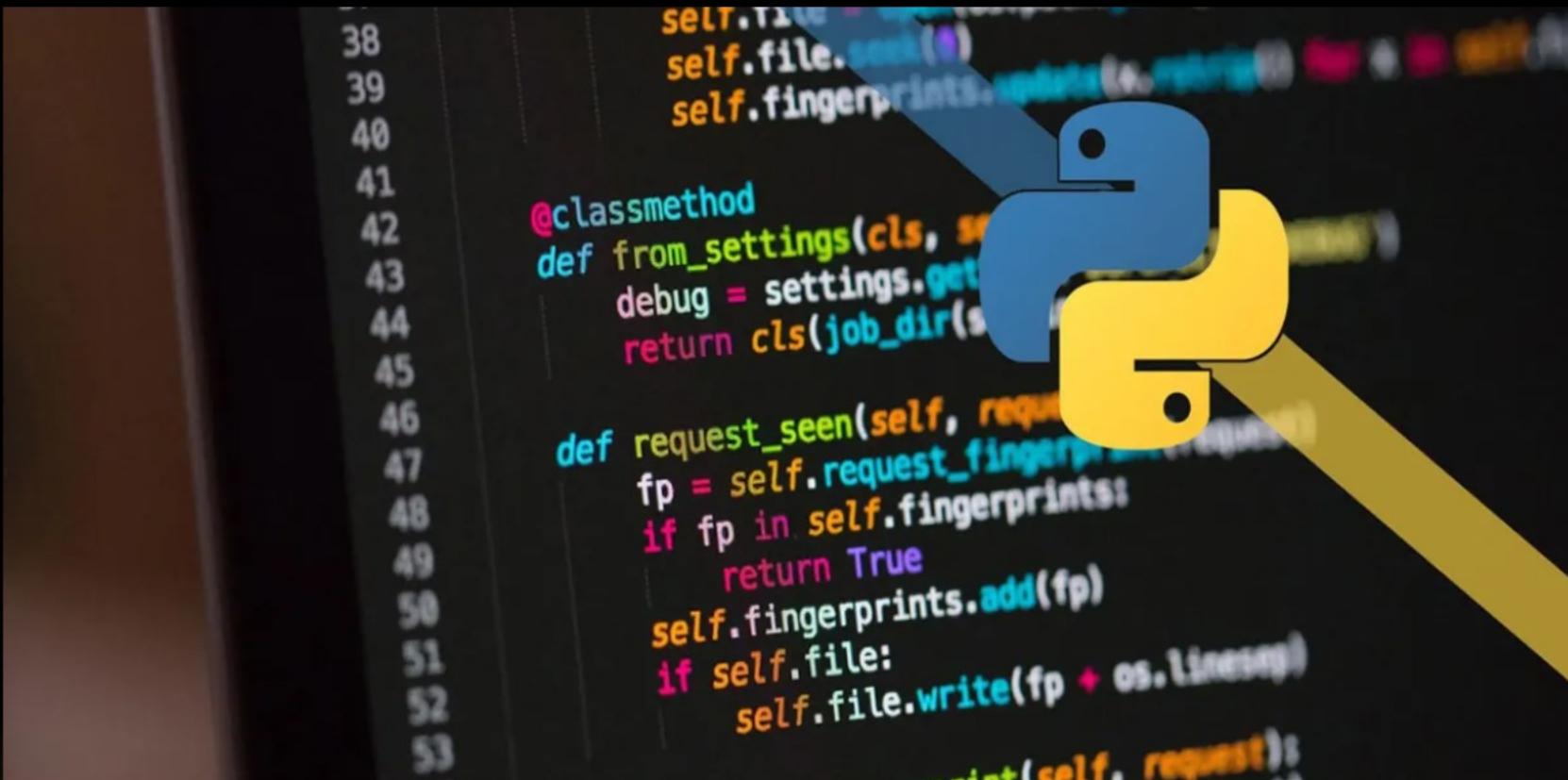
IskandarConqueror ADD - Se realizan variadas correcciones las vistas de reporte, se sep...

491bc6a · last week

History

Name	Last commit message	Last commit date
..		
pycache	ADD - Se agrega el modelo de report junto a su vista siguiendo lo ind...	last week
i18n	ADD - Se realizan variadas correcciones las vistas de reporte, se sep...	last week
models	ADD - Se realizan variadas correcciones las vistas de reporte, se sep...	last week
reports	ADD - Se realizan variadas correcciones las vistas de reporte, se sep...	last week
security	ADD - Se agrega el modelo de Motivo de Rechazo segun lo indicado en e...	last week
static/description	ADD - Se agregan pruebas del menu de configuracion, pruebas de verifi...	2 weeks ago
views	ADD - Se realizan variadas correcciones las vistas de reporte, se sep...	last week
_init__.py	ADD - Se agrega el modelo de report junto a su vista siguiendo lo ind...	last week
manifest.py	ADD - Se agrega al modulo su correspondiente traduccion, tambien se i...	last week

VEAMOS ALGUNOS MODELOS



The screenshot shows a dark-themed code editor interface with the title bar "Odoo 15".

Explorador (Left Panel):

- ODOO 15
 - addons\alkemy
 - clases
 - odoodegradein-main
 - gradein_services
 - _pycache_
 - i18n
 - models
 - _pycache_
 - _init_.py**
 - gradein_answer.py
 - gradein_equipment_type.py
 - gradein_image.py
 - gradein_order.py
 - gradein_question.py
 - gradein_reject_motive.py
 - reports
 - security
 - static
 - views
 - _init_.py
 - _manifest_.py**
 - README.md
 - requirements.txt
 - odoo-15.0
 - config
 - odoo.conf
 - Dockerfile
 - entrypoint.sh
 - requirements.txt
 - ESQUEMA
 - LÍNEA DE TIEMPO

Editor Area (Top Right):

addons > alkemy > odoodegradein-main > gradein_services > models > **_init_.py**

You, hace 7 días | 1 author (You)

```
1 from . import gradein_question
2 from . import gradein_equipment
3 from . import gradein_equipment_type
4 from . import gradein_answer
5 from . import gradein_image
6 from . import gradein_order
7 from . import gradein_reject_motive
```

You, la semana pasada • ADD - Se agrega el modelo de respuestas, así mism..

Bottom Tab Bar:

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS GITLENS COMENTARIOS

Terminal (Bottom Right):

```
main-IskandarConqueror-T1S2-141
main-IskandarConqueror-T1S2-147
main-IskandarConqueror-T1S2-150
main-IskandarConqueror-T1S2-151
main-IskandarConqueror-T1S2-155
* main-IskandarConqueror-T1S2-156
PS C:\Users\Usuario\Downloads\Acelerando Talento Fueguino\Odoo 15\addons\alkemy\odoodegradein-main\gradein_services> git push origin main-IskandarConqueror-T1S2-156
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (20/20), 3.66 KiB | 3.66 MiB/s, done.
Total 20 (delta 16), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (16/16), completed with 16 local objects.
To https://github.com/NQNseba/squad2_odoo_gradein.git
    f640158..8d4efd4  main-IskandarConqueror-T1S2-156 -> main-IskandarConqueror-T1S2-156
PS C:\Users\Usuario\Downloads\Acelerando Talento Fueguino\Odoo 15\addons\alkemy\odoodegradein-main\gradein_services>
```

ottoman starter Prettier

- **Modelo Odoo: answers / respuestas**
- **El modelo "respuestas" se ha creado para gestionar las respuestas en la aplicación. Incluye campos específicos que son relevantes para este propósito.**
- **Instalación**
- **Siga estos pasos para instalar y probar el módulo en su instancia de Odoo:**
 1. **Clone el repositorio.**
 2. **Copie la carpeta respuestas a la carpeta addons de su instancia de Odoo.**
 3. **Actualice la lista de módulos en Odoo.**
- **Criterios de Aceptación**
- **El modelo debe contener los campos necesarios para gestionar las respuestas.**
- **La instalación y activación del módulo en Odoo no deben generar errores.**

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons and a tree view of project files under 'EXPLORADOR'. The main area has a search bar at the top with the text 'Odoo 15'. A file named 'gradein_answer.py' is open in the center, showing Python code for a model named 'GradeInAnswer'. The code includes imports from 'odoo', class definitions, and field declarations. Below the code editor is a tab bar with 'PROBLEMAS' (1), 'SALIDA', 'CONSOLA DE DEPURACIÓN', 'TERMINAL' (selected), 'PUERTOS', 'GITLENS', and 'COMENTARIOS'. The terminal window displays a command-line session for pushing changes to a GitHub repository:

```
PS C:\Users\Usuario\Downloads\Acelerando Talento Fueguino\Odoo 15\addons\alkemy\odoo_gradein-main\gradein_services> git push origin main-IskandarConqueror-T1S2-156
main-IskandarConqueror-T1S2-141
main-IskandarConqueror-T1S2-147
main-IskandarConqueror-T1S2-150
main-IskandarConqueror-T1S2-151
main-IskandarConqueror-T1S2-155
* main-IskandarConqueror-T1S2-156
PS C:\Users\Usuario\Downloads\Acelerando Talento Fueguino\Odoo 15\addons\alkemy\odoo_gradein-main\gradein_services>
```

At the bottom, there are status indicators for tabs, file versions, and system information like Python version and file encoding.

Modelo: GradeIn Question / Preguntas

- **Descripción**
- **El modelo GradeIn Question se utiliza para representar preguntas en el sistema GradeIn. Cada pregunta tiene atributos como nombre, estado de activación y tipos de equipos asociados.**
- **Atributos del Modelo**
 1. **name: Campo de tipo Char que representa el nombre de la pregunta.**
 2. **active: Campo de tipo Boolean que indica si la pregunta está activa. El valor predeterminado es True.**
 3. **equipment_type_ids: Campo de tipo Many2many que establece una relación con los tipos de equipos en el modelo gradein.equipment.**

USO DEL MODELO

- Este modelo puede ser utilizado para gestionar preguntas en el contexto de Gradeln. Aquí hay algunos pasos básicos:
- Instalación del Módulo en Odoo:
- Copia este repositorio en la carpeta addons de tu instalación de Odoo.
- Actualiza la lista de módulos en Odoo para instalar el módulo.

VIEWS

```
factura.py x
mis_proyectos > factura.py > obtener_emisor
1 import xml.etree.ElementTree as ET
2
3 print("")
4 print("*"*65)
5 print("*"*20, f" Bienvenido Juan Magaña", "*"*20)
6 print("*"*65)
7
8 factura = {}
9
10 def obtener_todo(root):
11     for child in root:
12         print(f'{child.tag}: {child.attrib}')
13
14 def cargar_documento(path_xml):
15     # Carga el archivo XML
16     tree = ET.parse(path_xml)
17     return tree.getroot()
18
19 def obtener_emisor(root):
20     for child in root:
21         if 'Emisor'.lower() in child.tag.lower():
22             for atributo, valor in child.attrib.items():
23                 if atributo.lower() == 'Rfc'.lower():
24                     print(f" EfcEmisor: {valor}")
25                 elif atributo.lower() == 'Nombre'.lower():
26                     print(f" NombreEmisor: {valor}")
27                 elif atributo.lower() == 'RegimenFiscal'.lower():
28                     print(f" RegimenFiscal: {valor}")
```

File Edit Selection View Go Run ... ⏪ ⏩ gradein_services

EXPLORER

GRADEIN_SERVICES

- > __pycache__
- > i18n
- > models
- > reports
- > security
- > static
- > views
 - gradein_answer_views.xml
 - gradein_equipment_type_views.xml
 - gradein_equipment_views.xml
 - gradein_image_views.xml
 - gradein_order_views.xml
 - gradein_question_views.xml
 - gradein_reject_motive_views.xml
 - menu_gradein.xml
- __init__.py
- __manifest__.py

gradein_answer_views.xml

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
    <data>
        <!-- FORM -->
        <record id="view_gradein_answer_form" model="ir.ui.view">
            <field name="name">Grade In Answer Form</field>
            <field name="model">gradein.answer</field>
            <field name="arch" type="xml">
                <form>
                    <group>
                        <field name="name"/>
                        <field name="active"/>
                        <field name="price_reduction"/>
                    </group>
                </form>
            </field>
        </record>

        <!-- TREE -->
        <record id="view_gradein_answer_tree" model="ir.ui.view">
            <field name="name">Grade In Answer Tree</field>
            <field name="model">gradein.answer</field>
            <field name="arch" type="xml">
                <tree string="Answers">
                    <!-- Definimos las columnas que seran visibles -->
                    <field name="name"/>
                    <field name="price_reduction"/>
                </tree>
            </field>
        </record>
    </data>
</odoo>
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF XML Go Live

0 0 0 Connect -- NORMAL --

Windows Google Edge VS Code

15:59 15/12/2023

- Este código XML es una definición de vistas en el marco del framework Odoo. Odoo utiliza archivos XML para definir la estructura de las vistas en la interfaz de usuario, así como las acciones y modelos asociados. En este caso, la definición de vistas para un modelo llamado gradein.answer en Odoo.

Grade In Answer Action

[Create](#) Search...[Filters](#)[Group By](#)[Favorites](#)

Name

[Reduce the...](#)

- Aquí hay una explicación de cada parte del código:
- Formulario (`view_gradein_answer_form`):
- Este bloque define una vista de formulario para el modelo `gradein.answer`.
- Los campos que se muestran en el formulario son: `name`, `active`, y `price_reduction`.
- Está encapsulado dentro del campo (`<group>`) que organiza los campos visualmente.

Grade In Answer Action

Search...



Create



Filters

Group By

Favorites

1-1 / 1

 Name

Reduce the price by

 The equipment has corrupted software.

0.00

- Árbol (`view_gradein_answer_tree`):
 - Define una vista de árbol para el mismo modelo.
 - Los campos visibles en el árbol son: `name` y `price_reduction`.
 - La etiqueta `string="Answers"` establece el texto que se mostrará en la parte superior del árbol.
- Búsqueda (`view_gradein_answer_search`):
 - Define una vista de búsqueda para el modelo `gradein.answer`.
 - Incluye campos de búsqueda para `name` y `active`.
 - También establece dos filtros para `name` y `active` que afectarán la búsqueda y la agrupación de resultados.

Grade In Answer Action

[Create](#)[Archive](#)  

Search...

[Filters](#)[Group By](#)[Favorites](#)

1-1 / 1

 Name

Reduce the price by

0.00

true (1)

 The equipment has corrupted software.

0.00

Name

Archive

[Add Custom Group](#) ▾

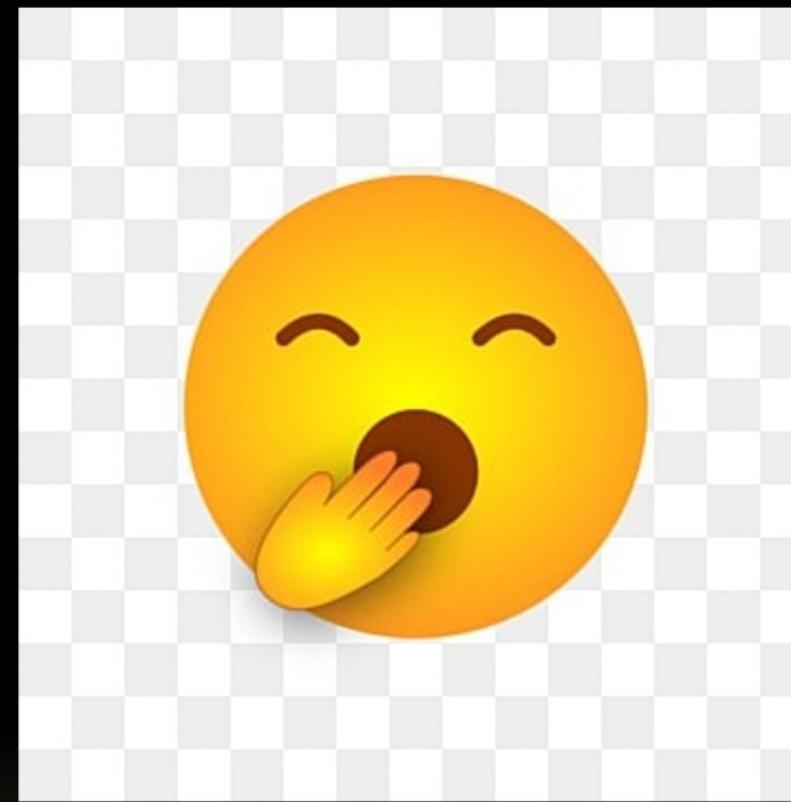
- **Acción (action_gradein_answer):**
- Define una acción de ventana para el modelo gradein.answer.
- La acción de ventana especifica cómo se abrirá la vista. En este caso, se define que la vista se abrirá en los modos tree, form, y search.
- Se establece que la vista predeterminada será la vista de árbol (view_gradein_answer_tree).
- En resumen, este código XML define las vistas (formulario, árbol y búsqueda) para el modelo gradein.answer en Odoo, así como una acción que utiliza la vista de árbol como predeterminada. Esto proporciona una interfaz de usuario para ver y editar registros de ese modelo.

PRESENTACIÓN EN VIVO

Módulos de Odoo

 Facturación	 CRM	 Comercio electrónico
 Ventas	 Proyecto	 MRP Mantenimiento
 Compra	 Inventario	 MRP
 Contabilidad	 Punto de venta	 Creador de sitios web

YA FALTA MENOS!!!!



DESAFÍOS EN EL DESARROLLO DEL CASO DE NEGOCIO: PEQUEÑA HISTORIA ILUSTRATIVA

- Estaba sentado frente a mi computadora, el código desplegado en la pantalla brillaba con una mezcla de esperanza y desesperación. Mi proyecto, un módulo para un e-commerce desarrollado con Python, Odoo, Docker y PostgreSQL, se resistía a funcionar correctamente. Mis ojos, fijos en la pantalla, veían líneas de código que parecían tener vida propia, pero no de la manera que yo quería. Intenté ejecutar el programa una y otra vez, pero cada vez la consola mostraba mensajes de error crípticos que me dejaban más confundido. Los sudores fríos comenzaron a aparecer mientras mi frustración se intensificaba. ¿Cómo podía ser que algo que debería ser tan simple se estuviera convirtiendo en un monstruo incontrolable de ceros y unos? Con cada intento fallido, mi paciencia se desvanecía, y mi mente se transformaba en una tormenta de 8 bits. Golpeé las teclas con fuerza, como si pudiera forzar a la máquina a entender mi desesperación. Pero el código persistía en su resistencia, como si se burlara de mis intentos. Lleno de frustración, me levanté de la silla y miré la pantalla con incredulidad. ¿Cómo había llegado a este punto? La tecnología que debería facilitar mi vida se había vuelto en mi contra, transformándome en una bestia digital de furia impotente.



- El Viaje hacia la Serenidad en 4K
- Después de tomar un momento para respirar y calmarme, decidí que necesitaba un enfoque diferente. No podía dejar que mi frustración me derrotara. Era hora de abrazar la paciencia y encontrar una solución. Me sumergí en la bibliografía, devoré vlogs y tutoriales relacionados a los componentes de mi proyecto. Busqué en foros, compartí mi código con la comunidad y expresé mis dudas. Poco a poco, el caos de 8 bits comenzó a ceder ante la serenidad de la resolución. La clave estaba en la paciencia y en la colaboración. Me di cuenta de que no estaba solo en este viaje; había otros desarrolladores que habían enfrentado problemas similares y estaban dispuestos a compartir su sabiduría. Compartí mi código en línea, describí mis problemas y recibí consejos valiosos que iluminaron el camino hacia la solución. Con el tiempo, las piezas del rompecabezas comenzaron a encajar. Los errores que antes parecían indescifrables ahora tenían sentido. La frustración se desvaneció a medida que la solución se materializaba frente a mí. Finalmente, el código se ejecutó sin problemas. La bestia de 8 bits se desvaneció, y en su lugar, la pantalla mostró el equilibrio de un estado Zen en 4K. La imagen ilustrativa, esta vez, reflejaba no solo mi victoria sobre el código, sino también la importancia de la paciencia, la colaboración y la perseverancia en el mundo del desarrollo de software.



CONCLUSIÓN

Este proyecto no solo nos proporcionó conocimientos técnicos sólidos en Odoo, Python, Docker y PostgreSQL, sino que también nos desafió a aplicar estos conocimientos en un entorno empresarial simulado. Los obstáculos superados durante este proceso han contribuido a nuestro crecimiento profesional y a la adquisición de habilidades valiosas para enfrentar proyectos empresariales complejos en el futuro. Este viaje de aprendizaje y desarrollo técnico ha sido fundamental para consolidar nuestro entendimiento y capacidad para abordar desafíos del mundo real en el campo del desarrollo empresarial y la tecnología.

Una mención especial a todos los que hicieron esto posible:
Gobierno de la Prov. De Tierra Del Fuego A.e.I.D.A.S. Arianna
Germiniani, Maria Piccino, Rodrigo Valladares, Bruno Cano,
Angel Guadarrama , Abel Aguirre Cabrera, a los compañeros
con los que terminamos esta capacitación el día de hoy y a
todos los que apoyaron esta iniciativa ¡desde ya Muchas
Gracias!

MUCHAS GRACIAS POR SU
ATENCION!

Somos Miguel y Sebastian
Squad 2