



CHECKSTOCK

I.E.S. Francisco Romero Vargas (Jerez de la Frontera)

Desarrollo de Aplicaciones Web

Curso: 2022/2024

Autor: Casandra Marín Angulo

Tabla de contenidos

01 Introducción

02 Finalidad

03 Objetivos

04 Investigación

05 Desarrollo

06 Problemas

07 Modificaciones

08 Mejoras

09 Conclusión

10 Bibliografía

01 Introducción

Este proyecto es parte del trabajo final de grado superior en desarrollo de aplicaciones web.

En esta presentación vamos a cubrir varios puntos:

- ✓ Finalidad del proyecto.
- ✓ Herramientas utilizadas.
- ✓ Objetivos.
- ✓ Dificultades encontradas.
- ✓ Desarrollo.
- ✓ Mejoras.

02 Finalidad

La elección de este proyecto surge por dos razones:

Experiencias vividas



Superación de dificultades

Finalidad del proyecto

Los propósitos a conseguir son:

1. Herramienta sencilla e intuitiva.
2. Buscar optimizar el control de inventario de las empresas.
3. Agilizar la gestión de inventario.
4. Ahorro de tiempo y recursos
5. Reducir errores en la gestión de existencias.
6. Anticiparse a la demanda.
7. Facilitar el trabajo manual a los trabajadores.



03 Objetivos

Para cumplir las acciones anteriores me propuse cumplir los siguientes objetivos:



1. **Seguimiento de inventario:** Controlar el stock de los artículos.
2. **Interfaz de usuario sencilla y clara.**
3. **Automatizar los procesos mediante avisos.**
4. **Unificación de la información en la empresa.**
5. **Restricción y control del usuario.**

Investigación: Competencia

04

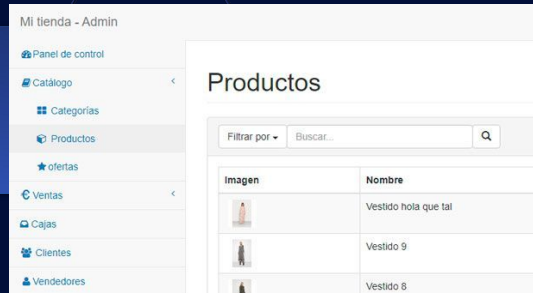


Antes de empezar con el proyecto hice una investigación sobre la competencia que tendría mi página con respecto a otras:

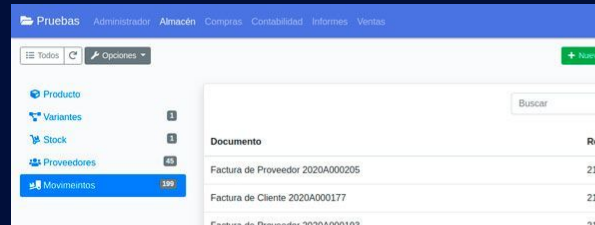
CATINFOG

FACTURASCRIPTS

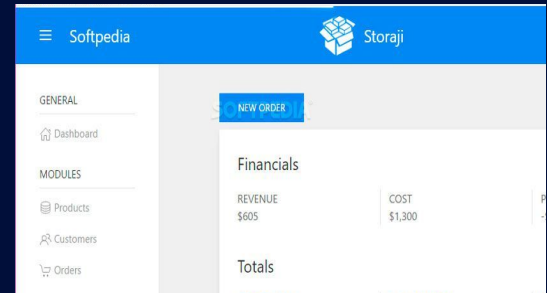
STORAJI



Controla Stock no
ubicación



Control de pedidos y cantidad de
Stock



Controla entradas y salidas

Con la investigación he llegado a la conclusión que puedo diferenciarme en:

- Saber dónde se encuentra el artículo.
- Controlar el stock de artículos en la tienda.
- Total de artículos por posibles errores.

Elegí la siguiente paleta de colores:



#2c3c54

Navegador
Fondo Contenedor
Botones hover



#D7D7D7

Fondo



#28a745

Botones



#ffffff

Input
Texto
Iconos



#731d25

Botón Warning
Texto Warning

Investigación:

Aplicaciones o bibliotecas

04



Aparte de lo aprendido en clase tuve que investigar varios apartados:

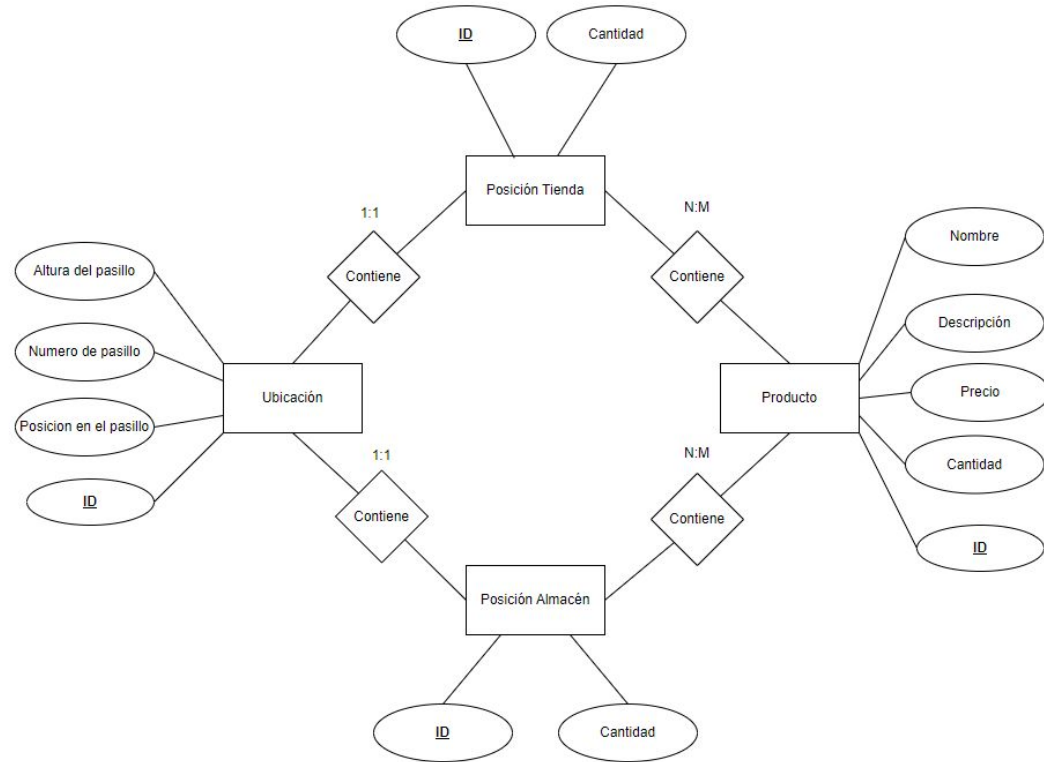
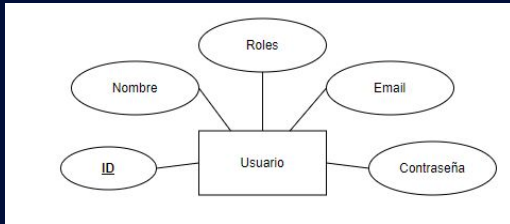
- Sabía que necesitaba introducir muchos productos, pero como ?
- Mejorar el manejo de la aplicación.
- Encontrar soluciones a los problemas que me planteaba el proyecto.
- Realizar restricciones de roles.



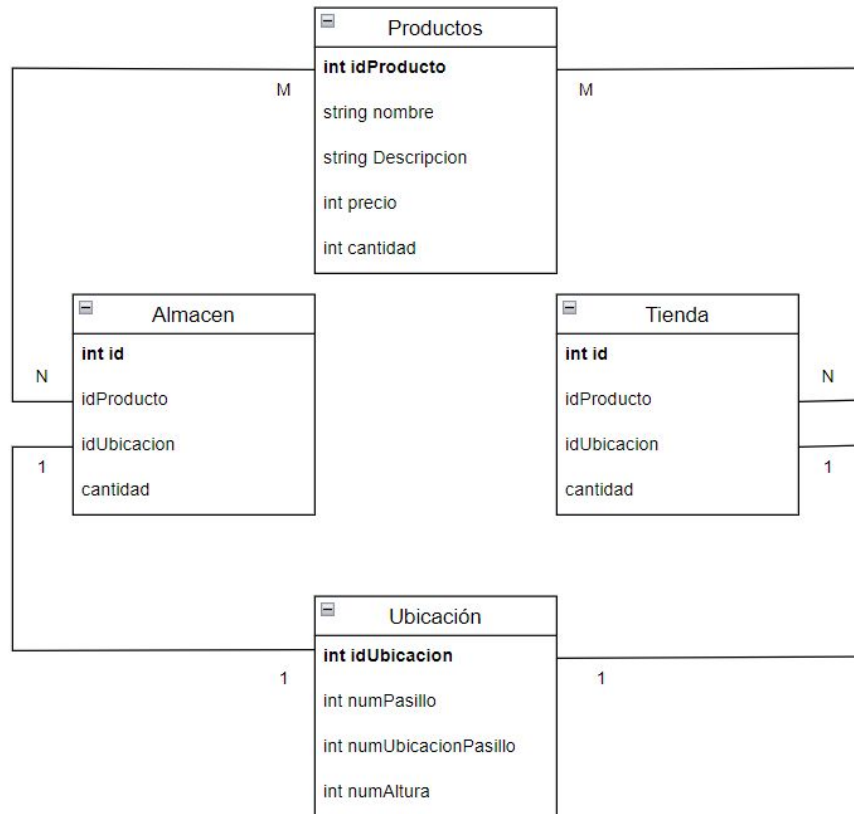
05 Desarrollo: Idea

Teniendo los objetivos claros , y la investigación necesitaba conocer que hacía falta para cumplir dichos objetivos :

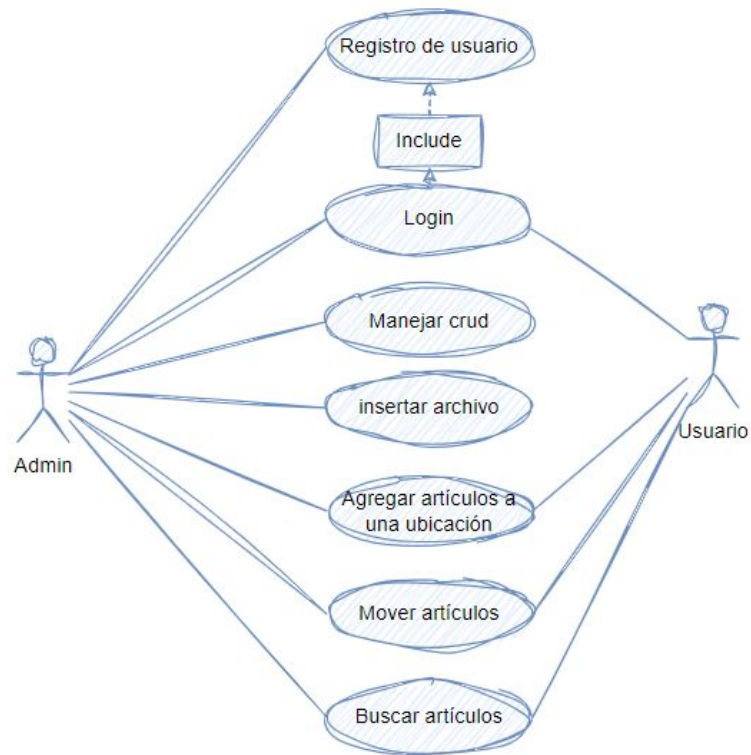
Entidades



Relaciones:



Seguridad:



05

Desarrollo: Instalación

Bibliotecas para Symfony:

- **composer create-project symfony/skeleton checkStock.**
- **composer require templates.**
- **composer require symfony/debug-pack.**
- **composer require symfony/orm-pack.**
- **composer require maker-bundle.**
- **composer require symfony/security-bundle**

05

Desarrollo: Realización del proyecto

- Creación de entidades.
- Creación de formularios.
- Creación del CRUD.
- Creación de los Controller.
- Creación de los templates.
- Cambios en security.yaml

```
access_control:  
  - { path: ^/admin, roles: ROLE_ADMIN }  
  - { path: ^/checkStock, roles: [ROLE_USER, ROLE_ADMIN] }  
  - { path: ^/artTienda, roles: [ROLE_USER, ROLE_ADMIN] }  
  - { path: ^/artAlmacen, roles: [ROLE_USER, ROLE_ADMIN] }  
  - { path: ^/moveArt, roles: [ROLE_USER, ROLE_ADMIN] }  
  - { path: ^/nuevos-articulos, roles: [ROLE_USER, ROLE_ADMIN] }
```

Problemas

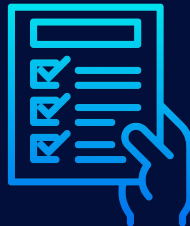
06



Instalación del CRUD



Formularios



Templates



Funciones

```
def add5(x):
    return x+5

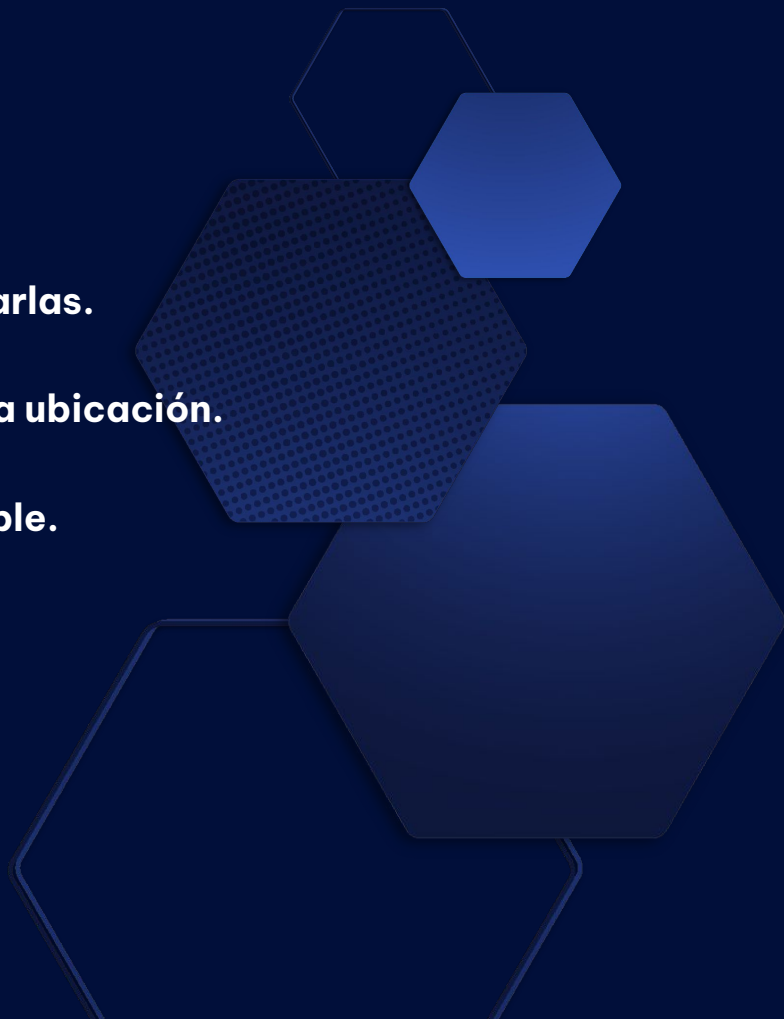
def dotwrite(ast):
    nodename = getNodeName()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '  %s [label=%s]' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print ' = %s';' % ast[1]
        else:
            print ']'
    else:
        print '[';
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print '  %s -> {' % nodename,
        for name in children:
            print '%s' % name,
```

07 Modificaciones

- ❌ Petición de los artículos en camino .
- ❌ Comprobar si se ha hecho la petición de los artículos.
- ✅ Tabla para artículos ingresados.
- ❌ Control del cambio realizado por el usuario.

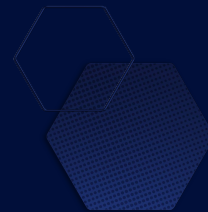
08 Mejoras

- ✓ Realizar peticiones de stock y mostrarlas.
- ✓ Controlar las cantidades totales en la ubicación.
- ✓ Hacer la aplicación más personalizable.
- ✓ Dar más opción a cambios posibles.
- ✓ Controlar el uso del usuario.

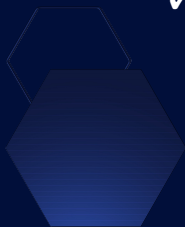


Conclusión

09



- ✓ Útil para empresas de todos los tamaños.
- ✓ Interfaz intuitiva.
- ✓ Posibles mejoras futuras.
- ✓ La seguridad es una prioridad.
- ✓ Acorta el tiempo de trabajo.



Bibliografía

- <https://catinfog.com/programas-almacen/>
- <https://www.rumpelstinski.es/actualidad/psicolog%C3%ADa-del-color-en-dise%C3%B1o-web>
- <https://symfony.com/doc/current/security.htmlhttps://symfony.com/doc/current/security.html>
- <https://symfonycasts.com/screencast/symfony-uploads/upload-in-form>
- <https://www.php.net/>
- <https://chatgpt.com/>
- <https://stackoverflow.com/>
- <https://app.diagrams.net/>
- <https://www.w3schools.com/>
- <https://fontawesome.com/icons>
- <https://getbootstrap.com/docs/>



Gracias!

¿Alguna Pregunta?

