Universidad Politécnica de Madrid

**Escuela Técnica Superior de Ingenieros Informáticos**

Grado en Ingeniería Informática

Propuesta de Trabajo Fin de Grado

**Aplicación Fiori App usando SAP Business Application Studio Venta de Productos por catálogo online**

**DELIVERY 1.0**

Autor: CRISTIAN RENE MEDINA ROJAS Tutor(a): [LARS-ÅKE FREDLUND](mailto:lfredlund@fi.upm.es)

Madrid, 20 de Julio del 2021

Contenido

1. [Introducción 3](#_bookmark0)
2. [Detalle de la Aplicación 4](#_bookmark1)
3. [Arquitecturas de la aplicación 5](#_bookmark2)
   1. [Arquitectura software (MVC) 5](#_bookmark3)
   2. [Arquitectura Cliente Servidor 5](#_bookmark4)
   3. [Arquitectura SAP Business Application Studio 6](#_bookmark5)
4. [Tecnología Utilizada 7](#_bookmark6)
   1. [Tecnología SAP 7](#_bookmark7)
   2. [SAP Cloud Plataform 7](#_bookmark8)
   3. [SAP Hana 7](#_bookmark9)
   4. [SAP Cloud foundry 8](#_bookmark10)
   5. [IDE de Desarrollo 9](#_bookmark11)
   6. [SAP UI5 9](#_bookmark12)
   7. [Servicio Work Flow 10](#_bookmark13)
   8. [Protocolo oData 10](#_bookmark14)
   9. [API REST 11](#_bookmark15)
5. [Desarrollo de la aplicación 11](#_bookmark16)
6. [Herramientas para el desarrollo 12](#_bookmark17)
7. [Planificación 13](#_bookmark18)
8. [Conclusiones 15](#_bookmark19)
9. Introducción

Hoy en día las ventas de artículos por internet utilizando aplicaciones desarrolladas en plataformas en la nube, se han vuelto muy populares. Por ende, pensar en el desarrollo de aplicaciones web supone una gran oportunidad para los desarrolladores y para los emprendedores poder sacar ventaja de estas tecnologías.

En el presente documento detallara el desarrollo de una aplicación utilizando la plataforma SAP Cloud Plataform. Utilizaremos las herramientas que SAP nos facilitan tanto los IDES de desarrollo como los entornos de desarrollo.

Se explicará en primera instancia los objetivos de la aplicación, seguidamente las herramientas necesarias para su desarrollo. A si mismo explicamos conceptos relevantes a entorno del mundo SAP, ya que considere importante mencionar estos conceptos. Finalmente haremos una planificación que llevaremos a cabo para el desarrollo del proyecto y un pequeño análisis en relación de las asignaturas que me han ayudado a obtener conocimientos para desarrollar mi aplicación.

1. Detalle de la Aplicación

**Delivery1.0 : Es una Aplicación cuyo objetivo es la venta de productos por catálogo por internet.**

**Delivery 1.0** es una aplicación cuyo objetivo es la venta de productos por internet. Su ciclo comienza en la compra de un producto y finaliza con la entrega del producto al usuario.

Observaremos la interacción de la aplicación.

* 1. El usuario ingresará a la aplicación y elegirá una serie de catálogos de

productos.

* 1. El usuario elegirá un producto y podrá ver su detalle como, por ejemplo:

precio, unidades en stock etc.

* 1. Una vez que se decida por el producto o productos, el usuario podrá añadir

al carrito de compra.

Por otro lado, en el carrito de compra podrá elegir el número de artículos del

producto seleccionado.

* 1. Finalmente, cuando el usuario se decida aceptar la orden de compra

tendremos una página de Detalles de la compra.

En la página detalle de la compra podrá ingresar los datos del usuario tanto su

nombre y su domicilio donde espera recibir su orden de compra.

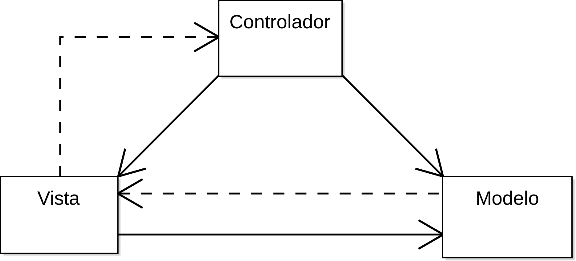
* 1. Al aceptar su compra se lanzará un servicio de Work Flow (ver Apartado 6.7) este servicio validará la orden de compra del usuario. El servicio de Work Flow es un servicio que requiere de una aprobación de carácter manual por una persona encargada de validar órdenes de compra. La persona encargada de validar se basará el algún criterio, como por ejemplo si el número de artículos solicitados por el usuario existen en el stock de la tienda. En cualquier caso, ya sea de aceptación o de rechazo se avisará al usuario toda la información de su pedido. Una vez que sea aprobada la orden de compra, la orden será entregada a un trasportista.
  2. Finalmente, la orden se entregará al usuario completando el objetivo el ciclo de la aplicación.

1. Arquitecturas de la aplicación

A continuación, muestro las arquitecturas que formaran parte de la aplicación

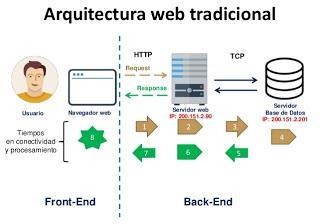
* 1. Arquitectura software (MVC)

Modelo Vista Controlador



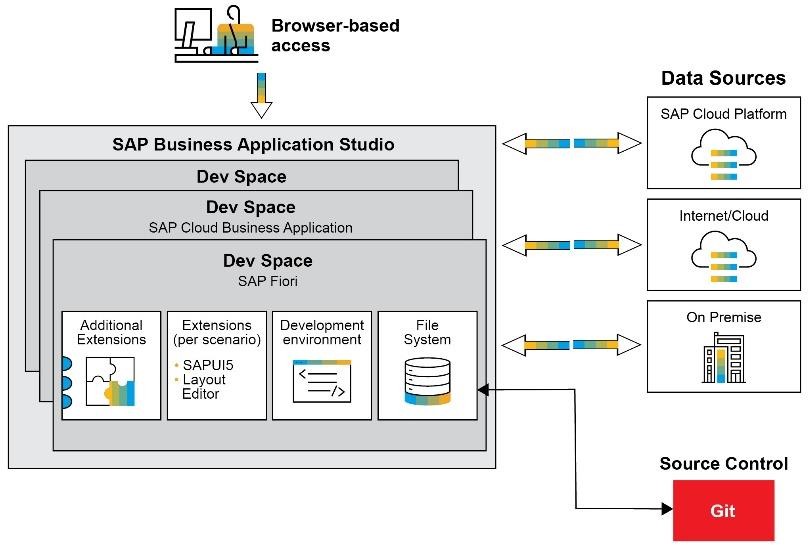
Para ello MVC propone la construcción de tres [componentes](https://es.wikipedia.org/wiki/Componente_de_software) distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) se basa en las ideas de [reutilización de](https://es.wikipedia.org/wiki/Reutilización_de_código) [código](https://es.wikipedia.org/wiki/Reutilización_de_código) y la [separación de conceptos](https://es.wikipedia.org/wiki/Separación_de_conceptos), características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

* 1. Arquitectura Cliente Servidor



La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados [servidores](https://es.wikipedia.org/wiki/Servidor), y los demandantes, llamados [clientes](https://es.wikipedia.org/wiki/Cliente_(informática)). Un cliente realiza peticiones a otro programa, el [servidor](https://es.wikipedia.org/wiki/Servidor), quien le da respuesta.

* 1. Arquitectura SAP Business Application Studio.



La Figura muestra la arquitectura y la filosofía básicas de SAP Business Application Studio. El desarrollador accede a la herramienta a través de un navegador y ejecuta uno o más espacios de desarrollo en paralelo. Cada espacio de desarrollo tiene un propósito determinado, el desarrollo de SAP Fiori, por ejemplo, y consta de herramientas relacionadas, extensiones y un sistema de archivos correspondiente para almacenar artefactos de desarrollo que se pueden versionar en un repositorio de Git adjunto. El acceso a fuentes de datos en la nube o en las instalaciones se habilita de la manera habitual mediante el consumo de servicios OData.

1. Tecnología Utilizada

# Dada que mi aplicación se basa en la tecnología SAP. A modo de resumen detallo conceptos relevantes para comprender el proceso de desarrollo de aplicaciones SAP Fiori.

En este presente apartado detallo algunos conceptos relevantes de lo que es el mundo SAP.

**Cloud Computing:** computación en nube. Servicios de software y de hardware

proporcionados por terceros a través de internet.

Primero vamos a centrarnos en un concepto relevante que es entrar en el mundo de SAP

* 1. Tecnología SAP

¿Qué es SAP SE?

SAP SE es una empresa [multinacional](https://es.wikipedia.org/wiki/Multinacional) [alemana](https://es.wikipedia.org/wiki/Alemana) dedicada al diseño de [productos](https://es.wikipedia.org/wiki/Software) [informáticos](https://es.wikipedia.org/wiki/Software) de gestión empresarial, tanto para empresas como para organizaciones y organismos públicos. En términos de ventas, SAP es la mayor compañía de software de [Europa](https://es.wikipedia.org/wiki/Europa) y la tercera más grande del mundo.

* 1. SAP Cloud Plataform

Que es SAP Cloud Plataform ?

SAP Cloud Platform es una [plataforma como servicio](https://es.wikipedia.org/wiki/Computación_en_la_nube) [desarrollada](https://es.wikipedia.org/wiki/Desarrollador_de_software) por [SAP SE](https://es.wikipedia.org/wiki/SAP_SE) **para la creación de nuevas** [**aplicaciones**](https://es.wikipedia.org/wiki/Aplicación_informática) o extender aplicaciones existentes en un entorno seguro de [computación en la nube](https://es.wikipedia.org/wiki/Computación_en_la_nube) gestionada por SAP.

* 1. SAP Hana

¿Qué es SAP HANA? (**Base de datos en la nube**)

SAP HANA es la implementación de [SAP SE](https://es.wikipedia.org/wiki/SAP_SE) de la tecnología de [base de datos](https://es.wikipedia.org/wiki/Base_de_datos) en memoria.

SAP HANA Aplicación en nube se refiere a la infraestructura basada en la [Computación en la nube](https://es.wikipedia.org/wiki/Computación_en_la_nube) para la entrega de aplicaciones (típicamente las aplicaciones existentes de SAP reescritas para ejecutarse en HANA).

**Entornos de desarrollo**

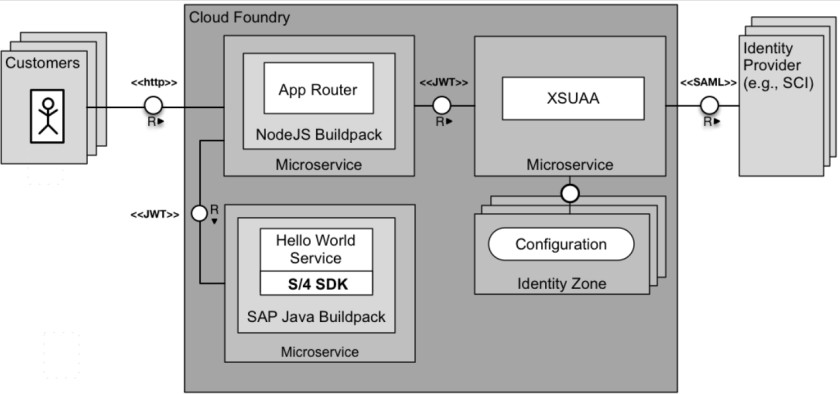
Desde el principio SAP Cloud Platform ha proporcionado dos entornos de desarrollo diferentes: Cloud Foundry y Neo, los cuales han estado evolucionando de manera paralela, trayendo novedades y mejoras continuamente.

Nosotros trabajaremos en el **entorno Cloud Foundry.**

* 1. SAP Cloud foundry

Que es SAP Cloud foundry ?

SAP Cloud foundry se basa en microservicios. La idea de cloud fondry es crear unas pequeñas cajas negras que interaccionan entre ellas a partir de APIs. Estas cajas negras o microservicios se pueden crear en cualquier lenguaje ya que lo importante es sus llamadas.



Cloud Foundry utiliza paquetes de compilación, que son API estandarizadas entre Platform-as-a-Service (PaaS) y las cargas de trabajo que se ejecutan sobre la capa de PaaS. Permiten que el entorno de Cloud Foundry admita una amplia gama de tiempos de ejecución y lenguajes de programación, que incluyen, por ejemplo, PHP, NodeJS, Ruby y Python, así como opciones de comunidad y traer su propio idioma.

El entorno de Cloud Foundry le permite elegir entre diferentes proveedores de IaaS y, por lo tanto, contribuye a la disponibilidad multicloud de SAP Cloud Platform.

* 1. IDE de Desarrollo

**SAP Business Application Studio** es un nuevo servicio de SAP Business Technology Platform (BTP) en SAP Cloud Foundry que ofrece un entorno de desarrollo moderno adaptado para el desarrollo eficiente de aplicaciones comerciales para SAP lntelligent Enterprise.

Disponible como un servicio en la nube, SAP Business Application Studio proporciona:

Una experiencia similar a la de un escritorio similar a los IDE líderes, con línea

de comandos y editores optimizados.

En el corazón de SAP Business Application Studio se encuentran los espacios de desarrollo, que son como máquinas virtuales aisladas en la nube que contienen herramientas personalizadas y tiempos de ejecución preinstalados por escenario empresarial, como SAP Fiori, extensiones de SAP S I 4HANA, Workflow, Mobile y más. Esto simplifica y ahorra tiempo en la configuración de su entorno de desarrollo y le permite desarrollar, probar, compilar y ejecutar de manera eficiente su solución" localmente o en la nube.

* 1. SAP UI5

¿Qué es SAPUI5 y para qué se utiliza?

[SAPUI5](https://sapui5.hana.ondemand.com/) es un framework de programación conformado por un conjunto de librerías de JavaScript para el desarrollo de aplicaciones basadas en HTML5. La principal característica de SAPUI5 es que las aplicaciones creadas cumplen con los 5 principios de SAP User Experience Strategy:

**SAPUI5 en SAP Cloud Platform**

SAPUI5 puede ser utilizado directamente en la nube mediante el entorno [SAP Cloud](https://cloudplatform.sap.com/index.html) [Platform](https://cloudplatform.sap.com/index.html). Este permite crear aplicaciones UI5 alojadas en la nube y, además, facilita el desarrollo mediante [SAP Web IDE](https://cloudplatform.sap.com/capabilities/product-info.SAP-Web-IDE.9e5c9d90-e8e0-4e82-aed2-09087a10c973.html).

* 1. Servicio Work Flow

Un Workflow es un **flujo de trabajo al que podemos incorporar actividades, operaciones, eventos, etc, dentro de nuestro entorno SAP**, bastante útil para cuando se necesita la iteración de varios usuarios dentro de un proceso. En nuestra aplicación en este servicio un responsable, validara la orden de compra .

* 1. Protocolo oData

El protocolo oData es utilizado por las aplicaciones SAPUI5 para realizar todas las transacciones de datos con el back-end ya sea a través del Gateway, XS Engine (con HANA) u oData Provisioning (con SCP).

Las principales características del protocolo oData:

Está basado en el protocolo de comunicación HTTP, por lo que sus recursos pueden

ser accedidos y editados con simples mensajes HTTP.

* + 1. Los recursos están definidos en un modelo abstracto de datos que está basado

en el Entity Data Model, que veremos más adelante.

* + 1. Utiliza URIs para la identificación de los recursos.
    2. Permite las operaciones CRUD (create, read, update, delete) sobre los recursos de una forma estandarizada.
    3. Utiliza los formatos de datos: XML y JSON.
    4. Cada servicio oData está representado por una URI, llamada la URI raíz del servicio. Existen dos documentos asociados al servicio oData que sirven para describirnos el servicio a través de llamadas GET HTTP:

# Para obtención de los datos utilizaremos la tecnología REST

* 1. API REST

¿Qué significa REST?

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

El formato más usado en la actualidad es el formato JSON, ya que es más ligero y legible en comparación al formato XML. Elegir uno será cuestión de la lógica y necesidades de cada proyecto.

REST se apoya en HTTP, los verbos que utiliza son exactamente los mismos, con ellos se puede hacer GET, POST, PUT y DELETE

1. Desarrollo de la aplicación

Se resume en los siguientes pasos el desarrollo de la aplicación.

Sin embargo, hay que destacar que esto es un resumen para una propuesta de TFG.

Este apartado se dotará de más información en la memoria final del Proyecto fin de

grado.

El desarrollo de la aplicación será el siguiente:

* 1. Diseño de la base de datos
  2. Implementación del servicio ODATA
  3. Diseño del mockup de la aplicación
  4. Desarrollo del mockup de la aplicación (interfaces graficas)
  5. Carga masiva de datos a la base de datos hana.
  6. Binding entre cliente y servidor
  7. Pruebas Unitarias
  8. Desarrollo de eventos de la aplicación
  9. Comprobación correcta de carga de datos
  10. Diseño de servicio de WorkFlow.
  11. Pruebas Unitarias
  12. Despliegue de aplicación en la nube
  13. Pruebas.

En la parte de pruebas se seguirán las pautas necesarias para hacer el testing de la

aplicación.

Se verificará:

La aplicación cumple con sus objetivos.

La aplicación cumple con los estándares basados en SAP.

Se realizarán las conexiones necesarias para el funcionamiento.

1. Herramientas para el desarrollo

Las herramientas necesarias para el desarrollo que utilizaré son: **Lenguajes de programación:**

Java Script, Java . **Base de Datos:**

Base de datos SAP HANA. **Repositorio:**

A modo de comunicación e interacción con mi tutor de TFG, la información será

almacenada en un repositorio en internet en específico GitHub.

Así mismo hay que mencionar que durante el desarrollo de la aplicación se irán investigando aplicaciones en internet que ayuden al desarrollo de la aplicación como, por ejemplo.

Aplicaciones basadas en planificación de proyectos etc.

1. Planificación

Con relación a la planificación a continuación, pondré los pasos que voy a desarrollar

para cumplir con el objetivo.

* El primer paso será definirme una planificación general de todo el proyecto.
* Hare unas estimaciones de duración y tiempo de finalización de cada tarea.
* El proyecto se llevará acabo de septiembre a diciembre. Los cálculos estimados son:

Número de créditos asignatura de TFG = 12 créditos Número de horas = 324

Numero de Semanas =15 semanas (septiembre- diciembre).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nº Tarea | Nombre Tarea | Semana | Fecha Inicio | Fecha Fin | Horas Estimación |
| 1 | * Redacción de Requisitos * Configuración de Software necesario para desarrollo de proyecto | Semana 1 | **Semana 1 Lunes** | **Semana 1 Viernes** | 24 horas |
| 1 | * Análisis de Requisitos * Modelo Entidad Relación. * Desarrollo memoria de TFG | Semana 2 | **Semana 2 Lunes** | **Semana 2 Viernes** | 24 horas |
| 2 | * Diseño de la base de datos * Comprobación de la base de datos * Diseño de   Mockup de aplicación | Semana 3 | **Semana 3 Lunes** | **Semana 3 Viernes** | 24horas |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | * Desarrollo de Mockup de aplicación * Pruebas Unitarias * Desarrollo memoria de TFG | Semana 4 | **Semana 4 Lunes** | **Semana 4 Viernes** | 24 horas |
| 4 | * Implementación servicio Work Flow * Implementación del Servicio * Despliegue de la base de datos en SAP Hana | Semana 5 | **Semana 2 Lunes** | **Semana 2 Viernes** | 24 horas |
| 5 | * Desarrollo de la funcionalidad de la aplicación * Desarrollo Memoria de TFG * Pruebas Unitarias | Semana 6,7,8,9 | **Semana 6 Lunes** | **Semana 9 Viernes** | 90 horas |
| 6 | * Desarrollo funcionalidad, eventos de la aplicación. * Comprobación de errores * Comprobación correcto Funcionamiento * Cambios alguna modificación de código. * Redacción de la memoria de TFG * Despliegue de Aplicación en la nube | Semana 10,11,12,13,14,15 | **Semana 6 Lunes** | **Semana 9 Viernes** | 114 horas |

Número Total de horas = 324 horas.

1. Conclusiones

# Relación del trabajo desarrollado con los estudios cursado

La presente aplicación pone en práctica todos los conocimientos alcanzados hasta el día de hoy, tanto la formación académica como en mi vida laboral.

Haciendo un análisis de los conocimientos que se necesitan para desarrollar mi aplicación puedo sacar las siguientes conclusiones:

El desarrollo de una aplicación necesita conocimientos de:

Programación en Java cuyo conocimiento se adquiere con las asignaturas de

programación: Programación 1, Programación 2, AED.

También requiere conocimientos de base de Datos y diseño de aplicaciones y planificación de proyectos cuyas asignaturas con base de Datos y Ingeniería del Software

Las aplicaciones web requieren información adquiridas en la Asignatura Sistemas Orientados a Servicios ya que pone en práctica todos los conocimientos como por ejemplo las llamadas de tipo REST

He detallado las asignaturas principales que durante la carrera he adquirido conocimientos princípiales para desarrollar la aplicación, sin embargo, quiero mencionar a todas las asignaturas en general de la carrera, ya que de ellas se aprende conceptos básicos que te sirven en primer o segundo plano para desarrollar la aplicación.