**SYSPRO EL TROGE WEB 1.0**

Descripción de la Arquitectura

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Propósito

# El presente documento fue creado con el fin de dar a conocer como se realizará la arquitectura del software SYSPRO EL TROGE WEB 1.0, asimismo justificaremos porque trabajamos con dicha arquitectura y cada uno de sus componentes.

# Objetivos y principios de la arquitectura

**Objetivos:**

Crear un modelo de sistema no muy complejo, para la realización de la arquitectura.

Adaptar las interfaces de presentación en un entorno web.

**Principios:**

Arquitectura de capas: objetivo principal es, separar las lógicas del negocio y de diseño con la finalidad de dar soporte a los procesos de negocio.

Se ha considerado una estructura de 3 capas: presentación, lógica del negocio y acceso a datos:

* Presentación: conformada por los contralores y formularios, en esta capa el usuario interactúa con el sistema.
* Lógica del negocio:
* Acceso a datos:

# Supuestos y dependencias

[List the assumptions and dependencies that drive architectural decisions. This could include sensitive or critical areas, dependencies on legacy interfaces, the skill and experience of the team, the availability of important resources, and so forth]

# Requisitos arquitectonicamente significativos

[Insert a reference or link to the requirements that must be implemented to realize the architecture.]

# Decisions, limitaciones y justificaciones.

[List the decisions that have been made regarding architectural approaches and the constraints being placed on the way that the developers build the system. These will serve as guidelines for defining architecturally significant parts of the system. Justify each decision or constraint so that developers understand the importance of building the system according to the context created by those decisions and constraints. This may include a list of DOs and DON’Ts to guide the developers in building the system.]

* Decision or constraint and justification
* Decision or constraint and justification

# Mecanismos de arquitectura

[List the architectural mechanisms and describe the current state of each one. Initially, each mechanism may be only name and a brief description. They will evolve until the mechanism is a collaboration or pattern that can be directly applied to some aspect of the design.]

## Architectural Mechanism 1

[Describe the purpose, attributes, and function of the architectural mechanism.]

## Architectural Mechanism 2

[Describe the purpose, attributes, and function of the architectural mechanism.]

# Anstracciones claves

[List and briefly describe the key abstractions of the system. This should be a relatively short list of the critical concepts that define the system. The key abstractions will usually translate to the initial analysis classes and important patterns.]

# Capas o marco arquitectonico

[Describe the architectural pattern that you will use or how the architecture will be consistent and uniform. This could be a simple reference to an existing or well-known architectural pattern, such as the Layer framework, a reference to a high-level model of the framework, or a description of how the major system components should be put together.]

# Vistas de arquitectura

[Describe the architectural views that you will use to describe the software architecture. This illustrates the different perspectives that you will make available to review and to document architectural decisions.]

## Recommended views

* **Logical:** Describes the structure and behavior of architecturally significant portions of the system. This might include the package structure, critical interfaces, important classes and subsystems, and the relationships between these elements. It also includes physical and logical views of persistent data, if persistence will be built into the system. This is a documented subset of the design.
* **Operational:** Describes the physical nodes of the system and the processes, threads, and components that run on those physical nodes. This view isn’t necessary if the system runs in a single process and thread.
* **Use case:** A list or diagram of the use cases that contain architecturally significant requirements.