
<Company Name>

<The liquor library>
Software Architecture Document

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

<Project Name>	Version: <1.0>
Software Architecture Document	Date: <01/06/19>
<Documento de arquitectura de software>	

Revision History

Date	Version	Description	Author
<01/06/19>	<1.0>	-	Bryan Naveja

<Project Name>	Version: <1.0>
Software Architecture Document	Date: <01/06/19>
<Documento de arquitectura de software>	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Architectural Representation	4
3.	Architectural Goals and Constraints	4
4.	Use-Case View	5
4.1	Use-Case Realizations	5
5.	Logical View	5
5.1	Overview	5
5.2	Architecturally Significant Design Packages	5
6.	Process View	5
7.	Deployment View	5
8.	Implementation View	5
8.1	Overview	6
8.2	Layers	6
9.	Data View (optional)	6
10.	Size and Performance	6
11.	Quality	6

<Project Name>	Version: <1.0>
Software Architecture Document	Date: <01/06/19>
<Documento de arquitectura de software>	

Software Architecture Document

1. Introduction

En el siguiente documento se plasmará la organización que se debe de tener en cuenta para la arquitectura del desarrollo del software.

1.1 Purpose

El propósito de este documento es estructurar y dar forma a todo lo que tiene que ver son la arquitectura del software a realizar ya que se incluirán todos los diagramas hechos y también se incluirán capturas de pantalla como evidencia.

1.2 Scope

El alcance de este documento tiene mucho que ver con el diseño de la arquitectura de software ya que se lograra actualizar de manera concreta todos los diagramas realizados para el software.

1.3 Definitions, Acronyms, and Abbreviations

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Architecture Document**. This information may be provided by reference to the project's Glossary.]*

1.4 References

*[This subsection provides a complete list of all documents referenced elsewhere in the **Software Architecture Document**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

1.5 Overview

A continuación, en este documento se 2organizará de manera sensata los siguientes puntos:

- 1- Representación de arquitectura
- 2- Logros de la arquitectura
- 3- Vista de casos de uso
- 4- Realización de casos de uso
- 5- Vista lógica
- 6- Vista de implementación
- 7- Vista de despliegue

2. Architectural Representation

En este apartado especificaremos los diagramas de casos de uso necesarios para el desarrollo del software:

<https://drive.google.com/file/d/1mPdbB1fu9fcQ2IfpKeMvQ6D5GHqNIDDF/view>.

3. Architectural Goals and Constraints

Para este apartado indicaremos los requisitos y objetivos del software que tienen algún impacto significativo al software.

Uno de los principales problemas que se tiene en este tipo de proyecto al ser un sistema de ventas es la seguridad del pago para el cliente.

<Project Name>	Version: <1.0>
Software Architecture Document	Date: <01/06/19>
<Documento de arquitectura de software>	

Para eso dentro de los casos de uso modelamos 2 tipos de pagos seguros.

4. Use-Case View

-

4.1 Use-Case Realizations

Para este caso de uso ilustraremos como funciona realmente el software dando las realizaciones de los casos de uso que seleccionamos.

<https://drive.google.com/file/d/1mPdbB1fu9fcQ2IfpKeMvQ6D5GHqNIDDF/view>

5. Logical View

En este apartado colocaremos las clases de importancia para el sistema o las que mas centralicen la utilización del mismo:

- Cliente
- Licor
- Conexión
- Reporte venta

Y para muestra dejamos aquí el enlace de la clase:

<https://drive.google.com/drive/u/1/folders/17ZzAwp-tqLXOfIcuZe-iWKeckiC3RiOy>

5.1 Overview

-

5.2 Architecturally Significant Design Packages

Para esta sección no tenemos en específico paquetes ya que todos los diagramas de clases fueron realizados y se comportan como solo un paquete.

6. Process View

-

7. Deployment View

En este apartado hablaremos un poco sobre dónde es que corre el software físicamente; por nuestra parte nuestro software corre sobre una computadora de escritorio la cual cuenta con conexión TCP/IP que es la que conecta cada uno de las computadoras al servidor principal y además tenemos una conexión USB 2.0 para la conexión de la impresora a cada una de las computadoras.

Esto se puede apreciar mucho mejor en el diagrama de despliegue el cual lo dejamos a continuación:

<https://drive.google.com/drive/u/1/folders/17ZzAwp-tqLXOfIcuZe-iWKeckiC3RiOy>

8. Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

<Project Name>	Version: <1.0>
Software Architecture Document	Date: <01/06/19>
<Documento de arquitectura de software>	

8.1 Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers.]

8.2 Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

9. Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

10. Size and Performance

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

11. Quality

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.]