



Facultad de Ingeniería  
Escuela de Ingeniería en Bioinformática  
Ingeniería de Software

---

# Título de informe

---

***Integrantes:***

José Rojas

Nicolás Romero

***Profesor:***

Felipe Besoain

***Ayudante:***

José Rizzo

# Índice

<b>1. Introducción</b>	<b>8</b>
1.1. Propósito . . . . .	8
1.2. Descripción breve del problema . . . . .	8
<b>2. Planificación del Trabajo</b>	<b>9</b>
2.1. Descripción del grupo de trabajo . . . . .	9
2.2. Estimación de esfuerzo . . . . .	10
2.2.1. Complejidad de Archivos . . . . .	10
2.2.2. Complejidad de las funcionalidades, cálculo de FP, grados de influencia y otros cálculos . . . . .	11
2.2.3. C.T Jones / COCOMO II . . . . .	13
2.3. Asignación de recursos . . . . .	16
2.4. Planificación temporal de actividades . . . . .	16
<b>3. Análisis</b>	<b>17</b>
3.1. Contexto . . . . .	17
3.1.1. Descripción General . . . . .	17

3.1.2.	Descripción de Clientes y Usuarios:	17
3.2.	Especificación de Requerimientos	17
3.2.1.	Funciones del Sistema	17
3.2.2.	Atributos del Sistema	17
3.2.3.	Atributos por Función	17
3.3.	Actores	18
3.4.	Casos de Uso	18
3.4.1.	Caso de Uso Esencial	18
3.4.2.	Diagrama de Caso de Uso	18
3.4.3.	Contrato	18
3.4.4.	Modelo Conceptual	18
3.4.5.	Diagrama de Secuencia o Colaboración	18
3.4.6.	Priorización	18
3.5.	Modelo de Dominio	18
3.5.1.	Entidades Reconocidas	18
3.5.2.	Modelo de Dominio	18
3.5.3.	Matriz de Rastreabilidad	18

<b>4. Validación</b>	<b>19</b>
4.1. Prototipo de validación funcional . . . . .	19
<b>5. Diseño</b>	<b>20</b>
5.1. Derivación del Modelo de Software . . . . .	20
5.1.1. Modelo de software inicial . . . . .	20
5.1.2. Diagramas de Clases . . . . .	20
5.1.3. Diagramas de Interacción . . . . .	20
5.1.4. Diagramas de Estados . . . . .	20
5.2. Refinamientos . . . . .	20
5.2.1. Lugar de Refinamiento . . . . .	20
5.2.2. Para cada Lugar . . . . .	20
5.2.2.1. Refinamientos considerados . . . . .	20
5.2.2.2. Selección y descripción de una opción . . . . .	20
<b>6. Implementación</b>	<b>21</b>
6.1. Código fuente completo (parcial) . . . . .	21
6.2. Modelo de implementación . . . . .	21

6.3. Dependencias . . . . .	21
<b>7. Anexos</b>	<b>22</b>
7.1. Glosario . . . . .	22

## Índice de figuras

## Índice de cuadros

# 1. Introducción

## 1.1. Propósito

## 1.2. Descripción breve del problema



## 2. Planificación del Trabajo

### 2.1. Descripción del grupo de trabajo

El grupo para este proyecto esta conformado por:

Nombre	Alias	Capacidades de profesión
José Rojas	JR	Programador Junior, Artista, Diseñador de Videojuegos
Nicolás Romero	NR	Programador Junior, Artista, Diseñador de Videojuegos

José Rojas : Programador Junior, Artista, Diseñador de Videojuegos.

Nicolás Romero : Programador Junior, Artista, Diseñador de Videojuegos.

Descripción del general del trabajo de JR: Se encargará de la gestión del proyecto, el plan del proyecto, trabajara en conjunto con su compañero para hacer los requerimientos, realizará tanto el diseño inicial como detallado de la app, hará la mitad de la codificación y por último hará la creación de la UI.

Descripción del general del trabajo de NR: Se encargará de parte de los requerimientos, trabajara en conjunto con su compañero para hacer el plan del proyecto, además realizará la prueba unitaria, prueba funcional, prueba de integración, prueba de la aplicación finalizada y por último, hará la mitad de la codificación del proyecto y la creación de Assets 3D.

## 2.2. Estimación de esfuerzo

### 2.2.1. Complejidad de Archivos

La siguiente tabla muestra las diferentes funciones necesaria para el correcto funcionamiento del software.

Tipos de parámetros	Nombres
EI (Entradas Externas)	Obtener Pieza Historica (PH), Buscar PH
EO (Salidas Externas)	Compartir Modelo 3d o Foto de PH en RRSS. Lista de piezas historicas obtenidas, Biblioteca de trofeos obtenidos.
EQ (Consultas Externas)	Visualizar PH, Rotar PH, Ver información de PH, Hacer Zoom in a PH, Hacer Zoom out a PH.
ILF (Ficheros lógicos internos)	Datos de PH, Datos de Trofeos.
EIF (Ficheros de interfaces externas)	Obtener piezas historicas compartidas en RRSS.

Objetos	Datos necesarios
Pieza Historica	Autor,Nombre de la pieza, Año de creación, Material usado, dimensiones, Ubicación en el museo, Nombre del Museo. Nombre del fotografo de la pieza. Detalles de la pieza y datos interesantes.
Trofeo	Tipo de Trofeo, Nombre de trofeo, detalles de trofeo.
Museo	Nombre, ubicación, fecha de construcción, lista de piezas historicas.

### 2.2.2. Complejidad de las funcionalidades, cálculo de FP, grados de influencia y otros cálculos

En las siguientes tabla se muestra la complejidad de las diferentes funcionalidades ligadas al software.

Entradas Externas EI	•	•	•
Nombre	DET	FTR	Complex
Obtener Pieza Historica	4	1	Low
Buscar Pieza Historica	2	1	Low
Obtener Trofeo	3	1	Low
Escanear Codigo QR del Museo	3	1	Low

Salidas Externas EO	•	•	•
Nombre	DET	FTR	Complex
Compartir Modelo 3d o Foto de PH en RRSS	3	2	Low
Biblioteca de piezas historicas obtenidas	3	1	Low
Biblioteca de trofeos obtenidos.	3	1	Low

Consultas Externas EQ	•	•	•
Nombre	DET	FTR	Complex
Visualizar Pieza Historica	2	1	Low
Rotar Pieza Historica	2	1	Low
Ver info. de Pieza Historica	2	1	Low
Zoom in a Pieza Historica	2	1	Low
Zoom out a Pieza Historica	2	1	Low
Ver info. de Museo	2	1	Low
Visualizar Museo	2	1	Low
Ver info. de Trofeo	2	1	Low

Ficheros lógicos internos ILF	•	•	•
Nombre	DET	FTR	Complex
Datos de Piezas Historicas	11	1	Low
Datos de Trofeos	4	1	Low
Datos de Museos	4	2	Low

EIF (Ficheros de interfaces externas)	•	•	•
Nombre	DET	FTR	Complex
Obtener piezas historicas compartidas en RRSS.	3	2	Low

Ahora podemos calcular los puntos de función sin ajustar, TUF<sub>P</sub>.

•	Bajo		Medio		Alto	
•	Número	Puntuación	Número	Puntuación	Número	Puntuación
Ficheros Lógicos Internos a la Aplicación	3	7	0	10	0	15
Ficheros Lógicos Externos a la Aplicación	1	5	0	7	0	10
Input Externos	4	3	0	4	0	6
Outputs Externos	3	4	0	5	0	7
Queries	7	3	0	4	0	6
•	•	71	•	0	•	0

El total de TUF<sub>P</sub>= 53 , cálculo y tabla generada a partir de la planilla de cálculo.

Grados de influencia y otros cálculos

Calculamos los grados de influencia a partir de la planilla de cálculo y podemos obtener lo siguiente:

TOTAL GRADOS DE INFLUENCIA	11
Puntos de Función Ajustados	53
VAF	0,76

### 2.2.3. C.T Jones / COCOMO II

Estimación de esfuerzo en base a C.T. Jones 1.996

FP	53
LOC en POO	20

C.T Jones	•
Métrica	Cálculo
Estimación de meses de desarrollo	4,894522709
Número de personas necesarias para el desarrollo (NP)	0,3533333333
Estimación del esfuerzo en personas/mes	1,729398024
Estimación de esfuerzo en horas hombre (160 horas x mes) 160 horas, ya que , se considera 1 jornada diaria de 8 horas	276,7036838

Estimación de esfuerzo en base a C.T. Jones 1.996 Software Estimating Rules of Thumb”  
y B. Boehm COCOMO II en fase de Early Design

COCOMO II	•
Métrica	Cálculo
Conversión de FP a LOC.	1060

Calculo de esfuerzo con COCOMO II con entrada de 1060 LOC y con factores de escala nominales.

Estimación	Optimista	Conservador	Pesimista
Meses de desarrollo	4,6	5,2	5,9
Personas necesarias	0,4	0,6	0,8
Personas/mes	2	3	4,5
Esfuerzo en horas hombre (160 horas x mes)	294,4	499,2	755,2

LOC Equivalentes	1060
------------------	------

Estimación del costo monetario en base al esfuerzo en HH conservador. Definiremos 2 perfiles:

Programador Junior: Perfil requerido para realizar la codificación crítica en base al diseño del software, gestión del proyecto, requerimientos, el plan del proyecto y los diseños base de la aplicación.

Artista Junior: Perfil requerido para la creación de assets 3d y UI para los diseños creados para el proyecto.

Estimación del coste de producir el software en 5.2 meses con 499 HH, bajo el modelo conservador de COCOMO II

Costo del producto	●	●	●	●
Etapa	Distribución en %	Distribución de T (HH)	Costo x Recurso	Recurso
Gestión del proyecto	3	14,97	Programador Junior	\$44.910
Requerimientos	8	39,92	Programador Junior	\$119.760
Plan del proyecto	1	4,99	Programador Junior	\$14.970
Diseño inicial	8	39,92	Programador Junior	\$119.760
Diseño detallado	8	39,92	Programador Junior	\$119.760
Codificación	48	239,52	Programador Junior	\$718.560
Documentación de usuario	1	4,99	Programador Junior	\$14.970
Prueba unitaria	3	14,97	Programador Junior	\$44.910
Prueba funcional	3	14,97	Programador Junior	\$44.910
Prueba de integración	4	19,96	Programador Junior	\$59.880
Prueba de la app	6	29,94	Programador Junior	\$89.820
Creación de Assets 3D	4	19,96	Artista Junior	\$39.920
Creación de UI	3	14,97	Artista Junior	\$29.940
●	100	499	●	\$1.392.210

**2.3. Asignación de recursos**

**2.4. Planificación temporal de actividades**



## **3. Análisis**

### **3.1. Contexto**

#### **3.1.1. Descripción General**

#### **3.1.2. Descripción de Clientes y Usuarios:**

### **3.2. Especificación de Requerimientos**

#### **3.2.1. Funciones del Sistema**

#### **3.2.2. Atributos del Sistema**

#### **3.2.3. Atributos por Función**

### **3.3. Actores**

### **3.4. Casos de Uso**

#### **3.4.1. Caso de Uso Esencial**

#### **3.4.2. Diagrama de Caso de Uso**

#### **3.4.3. Contrato**

#### **3.4.4. Modelo Conceptual**

#### **3.4.5. Diagrama de Secuencia o Colaboración**

#### **3.4.6. Priorización**

### **3.5. Modelo de Dominio**

#### **3.5.1. Entidades Reconocidas**

#### **3.5.2. Modelo de Dominio**

#### **3.5.3. Matriz de Rastreabilidad**

## 4. Validación

### 4.1. Prototipo de validación funcional

## 5. Diseño

### 5.1. Derivación del Modelo de Software

#### 5.1.1. Modelo de software inicial

#### 5.1.2. Diagramas de Clases

#### 5.1.3. Diagramas de Interacción

#### 5.1.4. Diagramas de Estados

### 5.2. Refinamientos

#### 5.2.1. Lugar de Refinamiento

#### 5.2.2. Para cada Lugar

##### 5.2.2.1. Refinamientos considerados

##### 5.2.2.2. Selección y descripción de una opción

## 6. Implementación

### 6.1. Código fuente completo (parcial)

### 6.2. Modelo de implementación

### 6.3. Dependencias

## **7. Anexos**

### **7.1. Glosario**