



Simulación de Sistemas

Trabajo Práctico Especial

Estrategia comercial de una  
software factory

Grupo 3

Civile, Juan Pablo	50453
Crespo, Alvaro	50758
Marseillan, Agustín	50134

17 de Junio de 2013

# Modelo Conceptual

## 1. Formulación del problema

El presente Trabajo Práctico busca representar la estrategia comercial de una consultora de software (*software factory*). En cuanto a la selección de proyectos, se busca la mejor combinación entre aprovechar el trabajo de los programadores y mantener cierta reserva de capacidad para no tener que rechazar proyectos *atractivos*, es decir, de mayor rentabilidad.

La simulación se basó en datos “pseudo-reales”, dada la dificultad que implicaba la recolección de datos y la escasez de ellos.

## 2. Modelado del problema

Se definieron las siguientes entidades:

- Proyecto
- Organización
- *Workflow*

Los proyectos fueron modelados como una *tupla* de horas-hombre, precio por hora, desarrolladores pretendidos, desarrolladores extra y fecha de entrega. Estas propiedades describen el tamaño, el ingreso que representa, la cantidad de desarrolladores que fueron negociados, los desarrolladores que fueron agregados al proyecto para cumplir la fecha y el tiempo máximo de desarrollo, respectivamente.

La organización representa la software factory en sí misma, y fue modelada como una entidad que posee un conjunto de desarrolladores, un *workflow*, una cola donde llegan los nuevos proyectos y una estrategia de aceptación de proyectos.

La entidad *workflow* está constituida por el conjunto de proyectos en desarrollo y la estrategia de asignación de recursos.

### 2.1. Restricciones - Límites del problema

Para el modelado del problema, se ignoraron algunas variables reales del problema, que hubieran hecho impracticable la representación y resolución del mismo. El simulador se ve restringido a un uso académico, dado que no tiene en cuenta muchas variables del mundo real.

Las variables que se ignoran son:

- La capacitación del personal y su curva de aprendizaje: se supone que el programador conoce el proyecto y no tarda en empezar a programar.

- Nuevos requerimientos en el proyecto: Desde un primer momento, se conoce el tiempo de desarrollo que involucrará un proyecto, no se pueden agregar funcionalidades a la mitad del desarrollo.
- Competencia, mercado, precios: estas variables están sujetas a la llegada de proyectos, pero no se toman como variables separadas sino que se engloban en la llegada de proyectos.
- La situación financiera de la empresa: se considera que la empresa tiene fondos como para pagar sus costos de funcionamiento durante la duración de la simulación.
- El traspaso de desarrolladores: se supone que la cantidad de desarrolladores permanece constante a lo largo de la simulación.

## 2.2. Variables de control

Se definen como variables de control del problema, las siguientes:

- Cantidad de programadores que se “reservan” (para proyectos *atractivos*).
- Estrategia de aceptación de proyectos.

## 3. Variables aleatorias

Se encontraron las siguientes variables aleatorias:

- Cantidad de proyectos que llegan en un determinado período.
- Tipo del proyecto (pequeño, mediano o grande).
- Tamaño del proyecto (medido en horas-hombre).
- Precio por hora del proyecto.
- Cantidad de desarrolladores pretendidos. Se considera que a la hora de negociar el proyecto, se acuerda un numero pretendido de desarrolladores asignados al proyecto.

## 4. Funciones objetivo

Las funciones objetivo que se consideran son:

- Costo de oportunidad, ingreso que hubieran generado los proyectos rechazados.
- Ingreso generado por los proyectos aceptados.
- Porcentaje de recursos utilizados.

Para representar los resultados y la evolución de estas magnitudes, se tiene una vista que muestra, paso por paso, estos valores y, al final de la simulación, un gráfico con los intervalos de valores.

Cabe destacar que para esto, previamente se debe fijar, para las estrategias de decisión que los tengan, los parámetros de entrada, buscando que sean los que arrojen los mejores resultados.

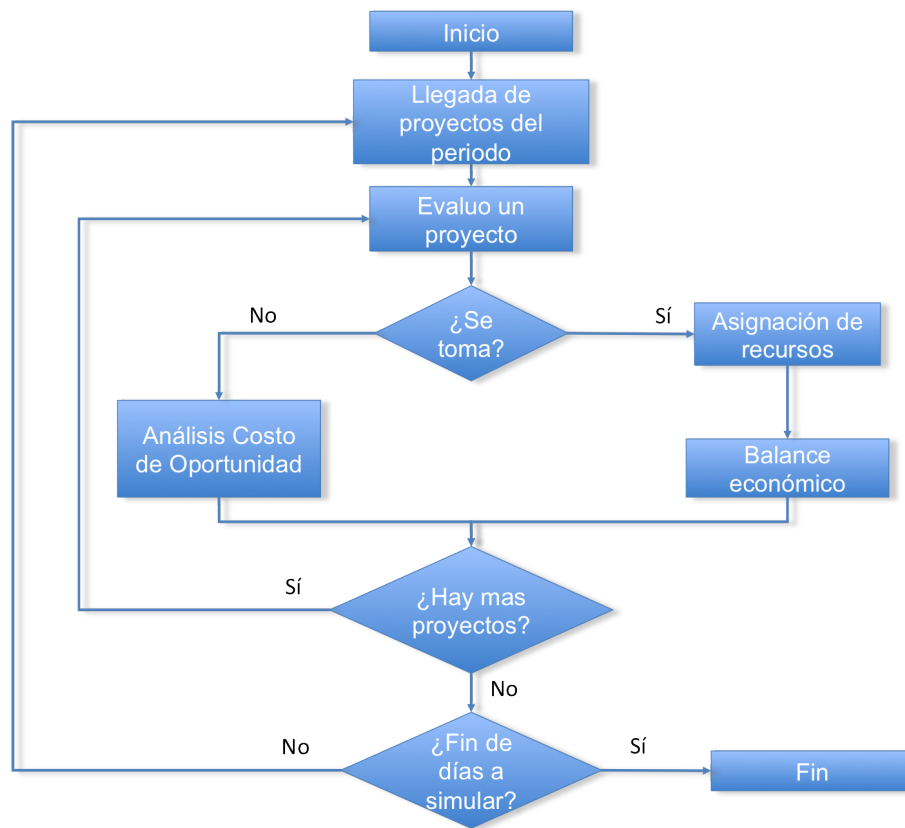
## 5. Plan de cuadros

El diagrama de cuadros, que se utilizará para llevar a cabo la simulación será el siguiente

<b>Estrategia</b>	<b>Devs Reservados</b>	<b>Ingreso</b> [ <i>min</i> , <i>mean</i> , <i>max</i> ]	<b>Costo de Oportunidad</b> [ <i>min</i> , <i>mean</i> , <i>max</i> ]	<b>% Recursos utilizados</b> [ <i>min</i> , <i>mean</i> , <i>max</i> ]
Estrategia 1 Precio HH	0	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	2	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	4	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	6	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
Estrategia 2	0	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	2	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	4	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]
	6	[..., ..., ...]	[..., ..., ...]	[..., ..., ...]

**Cuadro 1:** Plan de cuadros para la simulación

## 6. Diagrama de Bloques



**Figura 1:** Diagrama de bloque del simulador

## 7. Diagrama de Flujo

Muestras artificiales (MA)

Variables

1. Cantidad de Proyectos por mes(CA)  
VA Poisson ( $\lambda$ )
2. Tipo de Proyecto (TP)  
VA Uniforme
3. Tamaño de Proyecto [hs]  
VA Triangular(a, b, c)
4. Costo Hora-Hombre [\$]  
VA Triangular(a, b, c)
5. Desarrolladores pretendidos [hom-  
bres]  
VA Triangular(a, b, c)

- T: Tiempo del reloj de la simulación
- Tmax: Cantidad de tiempo a simu-  
lar
- P: Proyecto actual
- R: Recursos (programadores)
- M: Dinero
- CO: Costo de Oportunidad
- M(P): Ganancia del proyecto P
- R(P): Función de asignación de re-  
cursos usados por el proyecto P
- D(P): Decide si un proyecto es en-  
tregable en plazo considerando los  
proyectos que ya se tomaron
- O(P\_1, ..., P\_n): Ordena los proyec-

tos  $P_1$  a  $P_n$  según el criterio de la estrategia y marca al resultante  $P_1$  como atractivo

- A(P): Si la estrategia considera el

proyecto atractivo

- E(P): Desarrolladores reservados asignados al proyecto

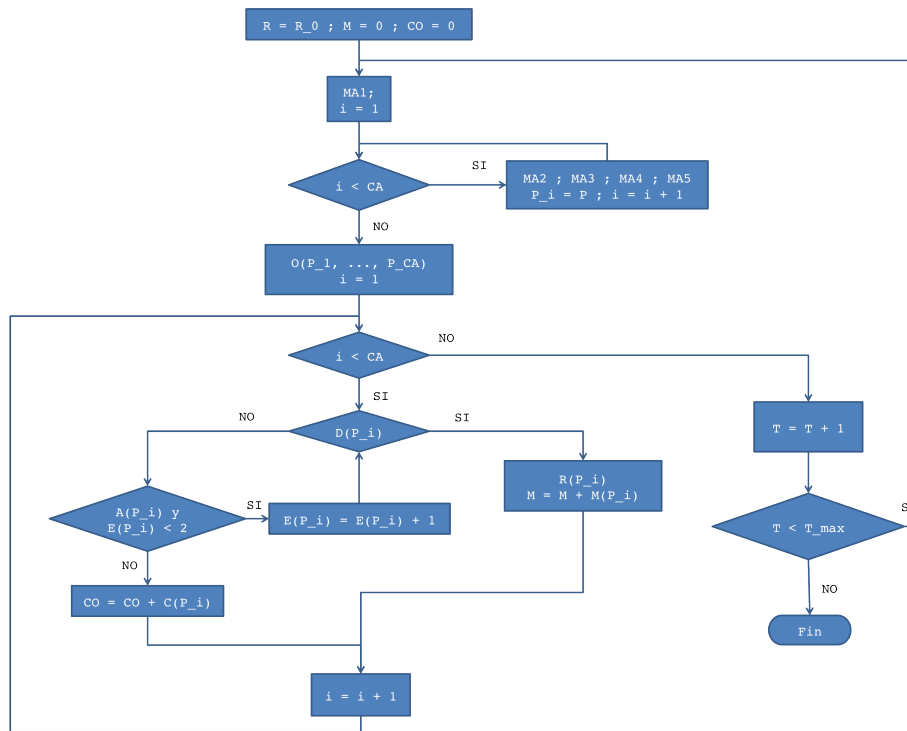


Figura 2: Diagrama de flujo del simulador

## 8. Plazo de entrega

Al momento de recibir un proyecto nuevo, se tiene la cantidad de horas hombre que este requiere y la cantidad de desarrolladores pretendidos para el proyecto. A partir de estas 2 variables se obtiene el plazo en meses del proyecto, haciendo el cociente entre las horas hombre y la cantidad de horas por mes que trabajarían los desarrolladores pretendidos.

## 9. Estrategias de decisión

Tanto en el diagrama de bloques como en el diagrama de flujo se observa una función o estrategia de decisión para aceptar o no un proyecto. A continuación se detallan las estrategias que se estudiarán. Debe ser tomado en cuenta que independientemente de la estrategia, si no se tienen suficientes recursos como para completar el proyecto antes de su fecha de entrega el valor de  $D(P)$ , es decir la función que acepta o no el proyecto, es siempre *NO*.

1. Se ordenan los proyectos arribados según el precio por hora, luego por la cantidad de horas del proyecto, y se aceptan, en caso de ser posible su desarrollo, en ese orden.
2. Se ordenan los proyectos encolados según su facturación (precio por hora \* horas-hombre), luego por el precio por hora, y se aceptan, en caso de ser posible, en ese orden.

## **10. Estrategia de asignación de recursos**

Para la asignación de recursos, al comienzo de cada período, se calcula la cantidad de horas/hombre que se van a asignar al proyecto de la siguiente manera: se toma el total de horas/hombre restantes del proyecto, llamémosle  $W$ , y se lo divide por la cantidad de períodos que faltan hasta la fecha entrega. Esto es la cantidad de horas que se le asignarán a ese proyecto en este período. En caso de queden recursos sin asignar, se asignarán al proyecto más cercano a su fecha de entrega. De esta forma, se busca balancear la asignación de recursos, buscando cumplir con las fechas de entrega, y en caso de ser posible, adelantar los proyectos cercanos a su fin, con el objetivo de liberar recursos para poder aceptar otros nuevos.

# Modelo de Datos

## 1. Variables Aleatorias

- A) Cantidad de proyectos por mes
- B) Tipo de proyecto
- C) Tamaño de proyecto [h]
- D) Precio hora-hombre del proyecto [\$]
- E) Desarrolladores Pretendidos del proyecto

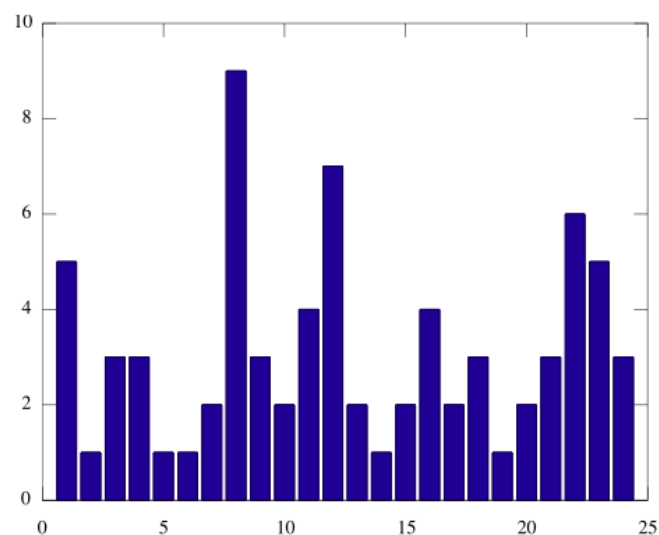
## 2. Estrategia de obtención

Para obtener las distribuciones de las variables aleatorias que utilizará el simulador, se tomaron muestras de los proyectos de algunas software factories. Luego se adaptaron a las distribuciones conocidas más similares, para simplificar el análisis y desarrollo del simulador. Los datos corresponden a los últimos 2 años.

A continuación se detallan las distribuciones de las distintas variables aleatorias.

## Llegada de proyectos

Para la generación de esta variable aleatoria se decidió utilizar una distribución de Poisson con media ( $\lambda$ ) 3. Se considera que esto aproxima correctamente la realidad.



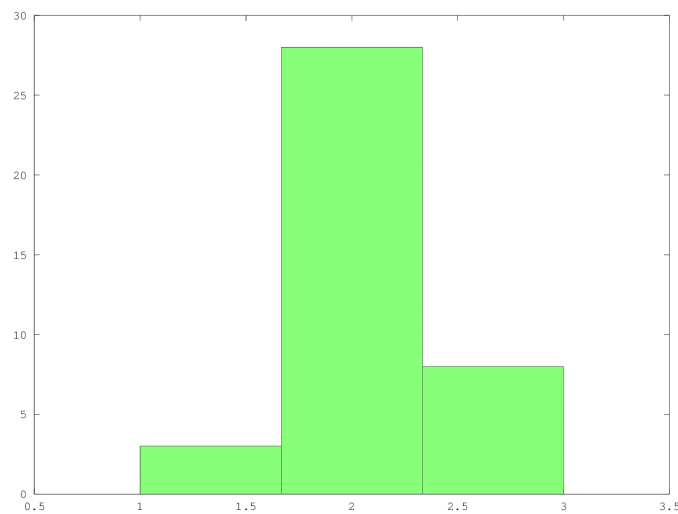
**Figura 3:** Gráfico de barras correspondiente a la llegada de proyectos.



## Tipo de proyecto

Para la variable aleatoria de tipo de proyecto, se estableció el uso de una distribución uniforme con las siguientes proporciones:

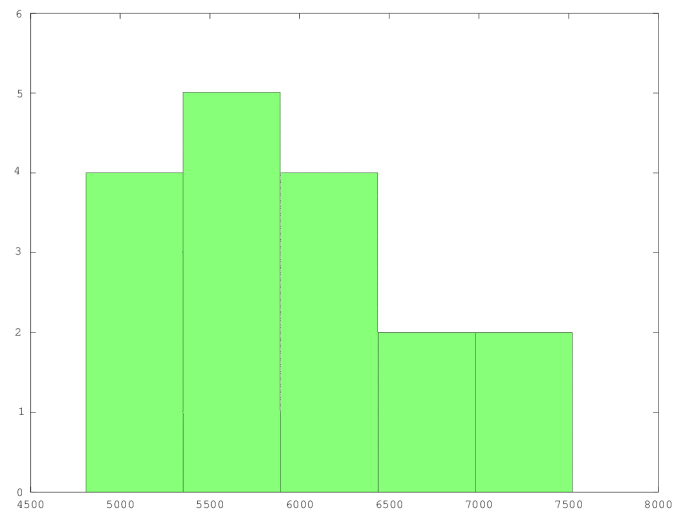
- Grande: 15 %
- Mediano: 75 %
- Pequeño: 10 %



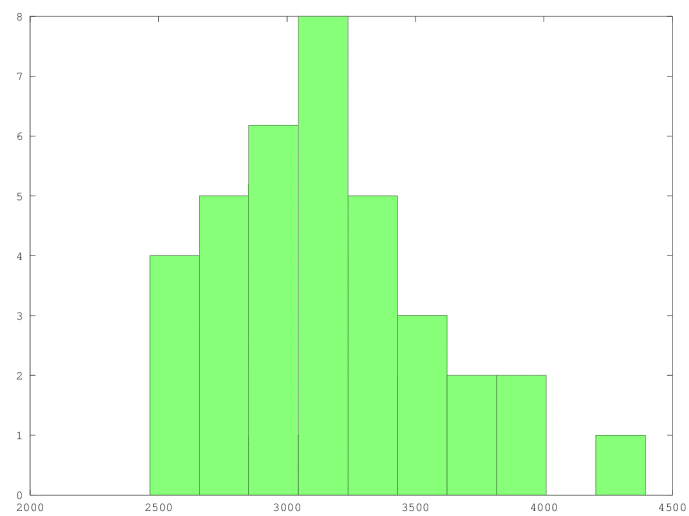
**Figura 4:** Histograma de los tipos de proyectos. El 1 representa un tamaño pequeño, el 2 mediano y el 3 grande.

## Horas Hombre de proyecto

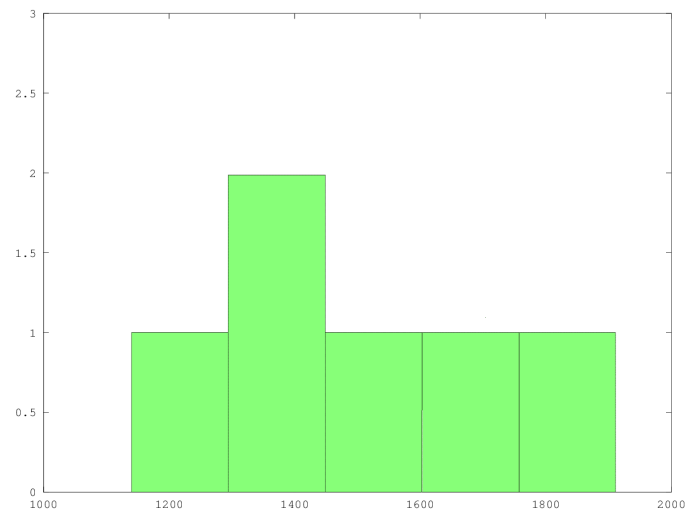
Dado que no se contaba con un gran número de muestras, y las distribuciones no podían ser fácilmente distinguidas, o no parecían seguir fielmente ninguna de las distribuciones más conocidas, se utilizó una distribución triangular, para aproximar según alguna “buena estimación”. Por esto, para la generación de la variable aleatoria se utilizó, según el tipo de proyecto sea pequeño, mediano o grande, una distribución triangular con parámetros (500, 2000, 1400), (2000, 4500, 3200) y (4500, 8000, 5300).



**Figura 5:** Histograma de las horas hombre de proyecto, para proyectos de tamaño grande.



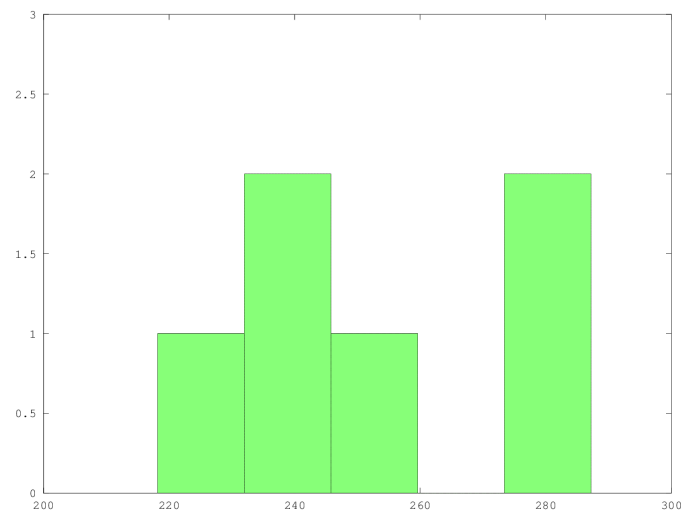
**Figura 6:** Histograma de las horas hombre de proyecto, para proyectos de tamaño mediano.



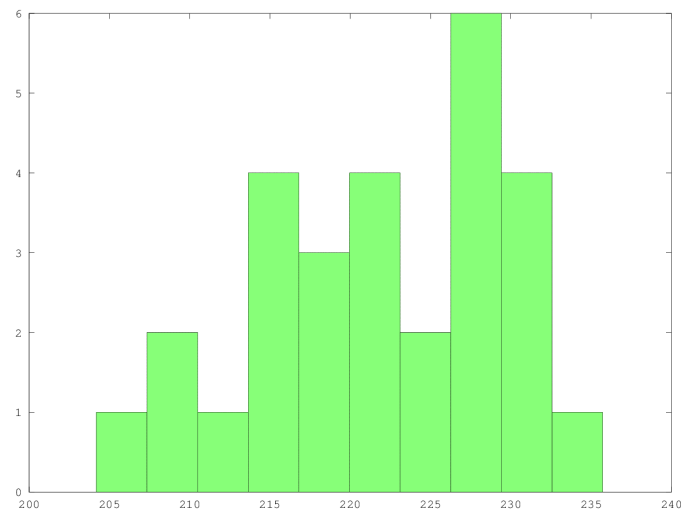
**Figura 7:** Histograma de las horas hombre de proyecto, para proyectos de tamaño pequeño.

## Precio por hora de proyecto

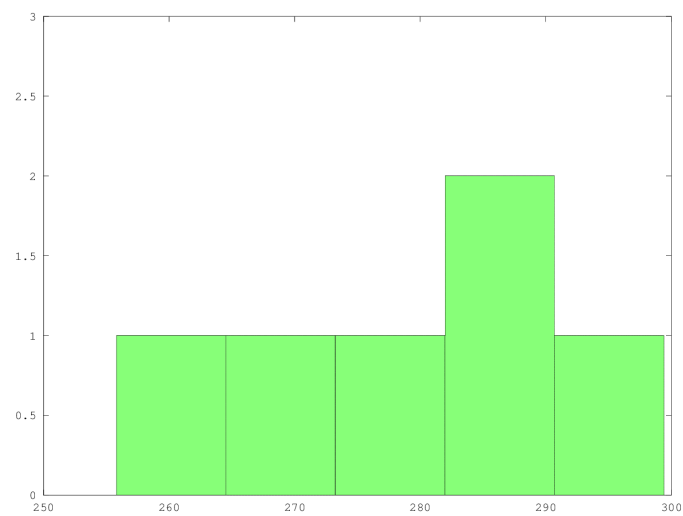
De nuevo, como se contaba con una gran cantidad de muestras, se utilizaron distribuciones triangulares para intentar aproximar dados algunos criterios estimativos. Para la generación de esta variable aleatoria se utilizó entonces una distribución triangular, variando sus parámetros según el tamaño del proyecto fuese pequeño, mediano o grande. Los valores de estos parámetros fueron (200, 310, 290), (200, 240, 220) y (180, 310, 230), respectivamente.



**Figura 8:** Histograma de precio por hora de proyecto, para proyectos de tamaño grande.



**Figura 9:** Histograma de precio por hora de proyecto, para proyectos de tamaño mediano.



**Figura 10:** Histograma de precio por hora de proyecto, para proyectos de tamaño pequeño.

## Desarrolladores Pretendidos de proyecto

Para la generación de esta variable, se estableció el uso de una distribución uniforme con las siguientes proporciones:

- Mínimo valor: 30 %
- Valor medio: 50 %
- Máximo valor: 20 %

Y, estos valores, varían según el tipo de proyecto, siempre manteniendo estas relaciones:

- Valor medio = Mínimo valor + 1
- Máximo valor = Mínimo valor + 2

y utilizando como valor mínimo para tipo de proyecto los siguientes valores:

- Grande: 5
- Mediano: 3
- Pequeño: 1

## Tabla de valores

A continuación se presentan los resultados de esta sección resumidos en 3 tablas de valores, con los parámetros de cada distribución.

Tipo	Porcentaje
Grande	15 %
Mediano	75 %
Pequeño	10 %

**Cuadro 2:** Cuadro de porcentajes de tipos de proyectos

	Tipo	Mín.	Máx.	Moda
Horas	Grande	4500	8000	5300
	Mediano	2000	4500	3200
	Pequeño	500	2000	3200
Precio	Grande	180	310	230
	Mediano	200	240	220
	Pequeño	200	300	290

**Cuadro 3:** Cuadro de valores para los parámetros de las distribuciones según tipo de proyecto

	Tipo	30 %	50 %	20 %
Devs Pretendidos	Grande	5	6	7
	Mediano	3	4	5
	Pequeño	1	2	3

**Cuadro 4:** Cuadro de valores para la distribución de desarrolladores pretendidos según tipo de proyecto

# Modelo Operacional

## 1. Arquitectura

Para el desarrollo del simulador se utilizará el lenguaje *Python*. Se hará uso de la librería *numpy* para la generación de números aleatorios, *matplotlib* para la generación de gráficos y *wxPython* para la GUI.

## 2. Etapas y Plan de Desarrollo

### **Etapas 1 - Modelo de Organización**

Como primer paso del desarrollo, se hará el modelo de organización y proyectos. Esto es, codificar las reglas de decisión y asignación de recursos. Desde luego que para poder desarrollar estas reglas, se debe codificar los tipos de datos proyecto y organización.

### **Etapas 2 - Motor de Simulación**

El principal objetivo de esta etapa es poder simular el análisis mensual de la organización. Con la llegada de proyectos, la decisión de toma de proyectos y la asignación de los recursos para el mes de trabajo.

### **Etapas 3 - Modelo de Datos**

El objetivo de esta etapa es ajustar de forma correcta el valor de todas las variables y parámetros especificados en el modelo de datos. Se deberían implementar los generadores de cada variable y deberían ser verificados.

### **Etapas 4 - Interfaz visual**

El objetivo de esta etapa es que pueda visualizarse la corrida de la simulación, mostrando las diferentes etapas, procesos y/o resultados intermedios. También se deberían poder visualizar las métricas en especial aquellas que son funciones objetivo.

## 3. Plan de experimentación

Para probar el simulador se diseñarán diferentes estrategias de aceptación y asignación de recursos, y se variará la cantidad de desarrolladores. Se probarán incluso escenarios de irreales, como la estrategia trivial de aceptar todo proyecto en según su orden de llegada, sólo con el motivo de probar el simulador y verificar que no surgen problemas.

## 4. Plan de pruebas

### 4.1. Verificación

- **Verificación de generadores aleatorios:** En el simulador se colocaron los histogramas de los valores que se generan durante la simulación. El objetivo de esto es verificar que en la simulación los histogramas se correspondan con las distribuciones de cada variable.
- **Verificación de estrategias de aceptación:** Una de las cosas importantes a verificar es que el simulador elija los caminos acorde a la estrategia seleccionada. Para ello se diseñaron dos pequeños conjuntos de testeo, compuestos por proyectos, generados correctamente según las distribuciones que corresponden, para los cuales, después de un corto tiempo de simulación se conocen los resultados.

# Análisis y recomendaciones

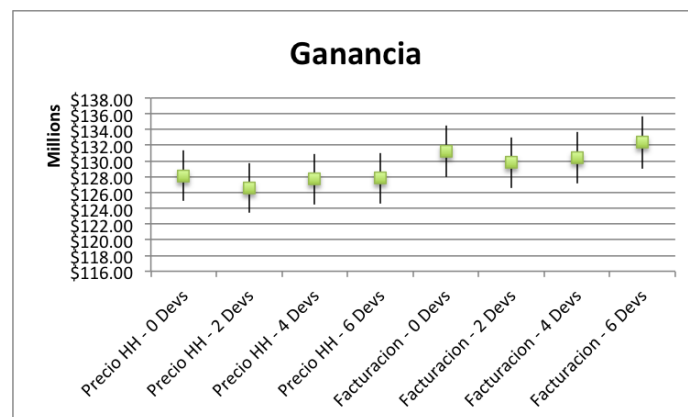
## 1. Resultados

Previo al cálculo de los resultados, se calculó el número  $N$  de corridas necesarias para obtener valores dentro del intervalo de confianza deseado. En particular se buscaba un intervalo del 5 % con un nivel de significación del 99 %. El cálculo del número de corridas arrojó que 100 corridas serían suficientes para lograr estos niveles de confianza.

A continuación se presentan el cuadro de resultados, y los gráficos de las funciones objetivos, para estrategia de aceptación de recursos. Cabe destacar que las estrategias están formadas por una combinación de 2 factores: la estrategia de aceptación en sí misma, y la cantidad de desarrolladores que se reservan para proyectos atractivos. Es por estos que los rótulos tienen la siguiente forma: “Nombre Estrategia - N devs”, como por ejemplo “Precio HH - 2 devs”, que se refiere a la combinación de la estrategia 1 con 2 desarrolladores reservados para proyectos atractivos.

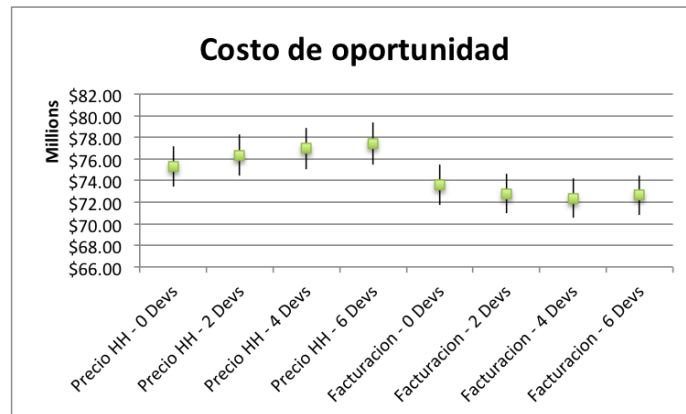
Estrategia	Devs Reservados	Ingreso [ <i>min</i> , <i>mean</i> , <i>max</i> ]	Costo de Oportunidad [ <i>min</i> , <i>mean</i> , <i>max</i> ]	% Recursos utilizados [ <i>min</i> , <i>mean</i> , <i>max</i> ]
Estrategia 1 Precio HH	0	[124 : 128 : 131]	[73 : 75 : 77]	[19,5 : 20 : 20]
	2	[123 : 126 : 129]	[74 : 76 : 78]	[19,5 : 20 : 20]
	4	[124 : 127 : 130]	[75 : 76 : 78]	[19,499 : 19,999 : 20]
	6	[124 : 127 : 131]	[75 : 77 : 79]	[19,493 : 19,993 : 20]
Estrategia 2 Facturación	0	[128 : 131 : 134]	[71 : 73 : 75]	[19,5 : 20 : 20]
	2	[126 : 130 : 133]	[70 : 72 : 74]	[19,5 : 20 : 20]
	4	[127 : 130 : 134]	[70 : 72 : 74]	[19,498 : 19,998 : 20]
	6	[129 : 132 : 136]	[70 : 72 : 74]	[19,5 : 20 : 20]

**Cuadro 5:** Resultados de la simulación

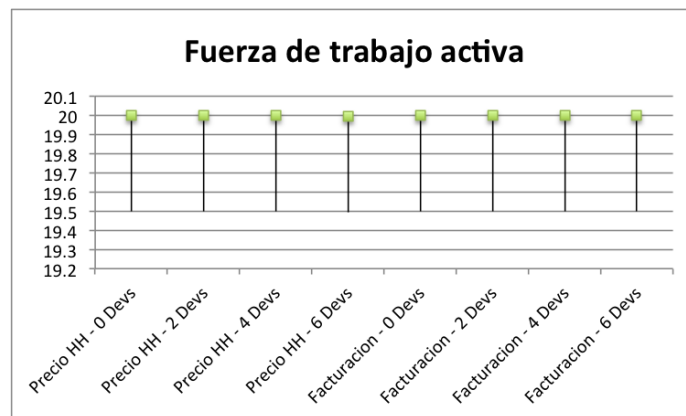


**Figura 11:** Ganancia con un intervalo del 5 % y 99 % de confianza





**Figura 12:** Costo de oportunidad con un intervalo del 5 % y 99 % de confianza



**Figura 13:** Fuerza de trabajo activa con un intervalo del 5 % y 99 % de confianza

## 2. Conclusiones

Un resultado que salta a la vista es que ambas estrategias logran un aprovechamiento óptimo o cuasi-óptimo de los recursos (ambas mantienen ocupados siempre o casi siempre a los 20 desarrolladores), por lo que no se puede basar solamente en este criterio para tomar una decisión acerca de que estrategia resulta más conveniente. Esto puede deberse también al diseño de la estrategia de asignación de recursos, la cual utiliza siempre a los desarrolladores que no tienen trabajo asignado para colaborar con los proyectos en desarrollo, pero dejándolos disponibles para aceptar nuevos proyectos.

Los resultados muestran que si se tiene en cuenta la ganancia, no hay diferencias significativas, aunque se podría aventurar que la estrategia 2, la que se base en la facturación, podría ser levemente mejor. Por otro lado, si se tienen en cuenta el costo de oportunidad, si se observa una clara diferencia en favor de la estrategia 2, que tiene

consistentemente un menor costo de oportunidad que la estrategia 1.

### **3. Recomendación**

En vista de los resultados y el análisis realizado, la recomendación sería que se utilice la estrategia 2, basándose en la facturación de los proyectos para su aceptación. Queda claro que esto se da en un marco muy simplificado, con variables que fueron excluidas y aspectos que fueron desestimados para facilitar el desarrollo y el análisis de los datos.

### **4. Próximos pasos**

Algo interesante para agregar al simulador sería la inclusión de más estrategias de decisión, algunas más triviales y algunas más complejas, para comparar con las actuales y ver si realmente son tan buenas como parecen.

Además, en algunos casos parecería que tener una mayor cantidad de desarrolladores reservados podría hacer que alguna estrategia fuera mejor, con lo cual sería interesante extender los rangos de valores para dicha variable de control y encontrar los puntos de quiebre para cada estrategia.

Otra mejora podría ser la consideración de crisis o períodos caóticos, en los que la demanda de proyectos se reduce notablemente o incluso se anula. Esto reflejaría las condiciones de un mercado como el argentino, que se caracteriza por su inestabilidad, y mostraría como las estrategias se comportan ante la falta de trabajo asegurado.