# Testing I

# **Examen integrador**

Les pedimos que lean atentamente las siguientes consignas y respondan a las preguntas de acuerdo a lo solicitado.

**No se aceptarán links de Drive, solo documentos adjuntos. Caso contrario, el examen no será considerado para su corrección.**

Nota aclaratoria: al enviar el formulario con el adjunto se debe esperar la confirmación del profesor **antes de salir de la sala de Zoom** para garantizar que se recibió correctamente para posterior corrección. Caso contrario, no se recibirá la evaluación y el alumno deberá recuperar esta instancia de evaluación. **Solo se recibirá 1 (un) documento por alumno**.

**Duración**: 1 hora 30 minutos.

**Nombre y Apellido:** Patricio Alloco

Parte teórica

1. Explique con sus palabras el concepto de Prueba Funcional.

La prueba funcional se hace para probar las funcionalidades de sistema. Nos enfocamos en que cumpla bien las funciones y en el que hace el sistema. Con estas pruebas vas probando como se comporta el software.

1. ¿Qué técnica de prueba considera correcta para aplicar en una aplicación con muchos menús y workflows?

Justifique su respuesta.

Yo usaría una técnica de transición de estados porque de esta forma voy evaluando el comportamiento y los cambios de acuerda a cada camino de los distintos menus que voy evaluando.

1. Mencionar las principales diferencias entre **debugging** y **testing**.

El debugging consiste en analizar un software o una parte de el buscando los posibles fallos para removerlos. Se suele hacer linea por linea y para ello de utilizan breckpoints que te ayudan a ir frenando la ejecución para poder analizar valores en determinado momento. Por su parte el testing es mas generas porque a parte de las funciones de debaguear parte del código va mas alla con distintas pruebas y planteando distintos escenarios para encontrar distintos tipos de fallos.

1. Mencionar que pruebas se realizan en el ambiente/entorno de **Stage-STG**

En este ambiente el objetivo es emular el entorno de producción. Se realizan pruebas de mantenimiento, de seguridad, de rendimiento, de infraestructura, de carga, estrés y escalabilidad, de gestión de memoria, compatibilidad e interoperabilidad y de migración de datos.

1. Indicar cuales son los objetivos específicos de las pruebas de aceptación (Niveles)

El objetivo específico es evaluar el comportamiento y las capacidades de todo un sistema. A partir de ahí se puede evaluar el grado de preparación que tiene el sistema para su despliegue y uso por parte de los clientes

# **Requerimientos Digital Booking**

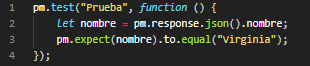
**Digital Booking** es nuestro sitio de reservas: [**Ingresá desde acá**](http://fe.deitech.online/)

**Requerimientos:**

* El sistema deberá permitir el registro de un tipo de usuario “cliente”.
* Los datos necesarios serán: nombre, apellido, e-mail y contraseña.
* Se debe validar que todos los campos estén completos y tengan el formato correcto:
  + Nombre (de type=”text”)
  + Apellido (de type=”text”)
  + Email (de type=”email”)
  + Password (de type=”password”- min:6 caracteres)
  + Repetir password (de type=”password”- min:6 caracteres)
* El sistema deberá permitir loguear a los usuarios. En caso de que el e-mail o la contraseña sean incorrectos, se debe mostrar un mensaje de error.
* El sistema deberá permitir a los usuarios realizar una reserva. Los usuarios registrados pero no habilitados no pueden realizar reservas.
* Se habilitará un usuario con el link que llega al e-mail de registro.
* Al momento de realizar una reserva, el sistema deberá enviar un mail al usuario logueado con toda la información detallada de su reserva.
* El sistema deberá permitir a los usuarios realizar búsquedas de alojamientos según fecha y lugar.
* Los lugares se mostrarán mediante un listado y debe poder filtrar por palabras claves, mientras que la fecha no debe permitir seleccionar una fecha anterior a la actual y la estadía mínima permitida debe ser de 1 noche.
* En el caso de que no existan alojamientos disponibles para las fechas deseadas, el sistema deberá indicarle al usuario con un mensaje.
* El sistema permitirá la habilitación del usuario mediante el envío de un SMS.
* El sistema deberá permitir agrupar los alojamientos por tipo y también mostrar los más recomendados por los clientes.
* El sistema deberá permitir al usuario agregar alojamientos a favoritos, como así también, quitarlos.
* El sistema deberá permitir al usuario ver las reservas que realizó con los detalles de la misma.
* La aplicación debe ser responsive, es decir, debe adaptarse a las diferentes resoluciones del navegador hasta llegar a la versión móvil.
* La aplicación debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

Parte práctica

1. ¿Qué nos permite validar el siguiente **test de Postman**?



Este es un test de JSON value check. Sirve para validar si el campo nombre del objeto de la respuesta es igual a Virginia.

1. Identificar una petición GET y una POST de nuestra app [**Digital Booking!**](http://fe.deitech.online/), (Recomendación: Utilizar la herramienta de desarrollo > DevTools) Explicar brevemente de qué se trata una petición POST.

**Respuesta pegada desde postman abajo de todo. Lo pegue 2 veces una con el fondo de visual y otra no por las dudas.**

1. Redactar brevemente **1 caso de prueba** **negativo** que aplicarías en la página de [**Digital Booking!**](http://fe.deitech.online/) (**No se requiere escribirlos en formato de template**).

Intente crear un nuevo usuario con todos los campos del formulario de registro vacíos y la app no me dejo continuar sin antes llenar los campos requeridos.

1. Mencionar **1 defecto** que encuentres en la página de [**Digital Booking!**](http://fe.deitech.online/) (**No se requiere escribirlos en formato de template**).

El defecto que encuentro es que una vez hecha la reserva no tengo opción de modificarla o eliminarla.

1. Si estoy trabajando con **Jest** y quiero validar que el resultado devuelto sea **false**. ¿Qué **matcher** puedo utilizar? Dar un ejemplo de un posible test para cualquier sistema bajo prueba

Lo que deberíamos usar es el matcher **.toBeFalsy**

test(‘compruebo que el empleado este dado de baja’, () =>{

const isFalse = bajaEmpleados(id);

expect(isFalse.altaEmpleado).toBeFalsy();

}

Respuesta 7:

{

    "info": {

        "\_postman\_id": "6bd3d9db-6f1c-4d80-aafc-0bd6f14d86f2",

        "name": "Final Testing BookingApp - Patricio Alloco",

        "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

        "\_exporter\_id": "18541597"

    },

    "item": [

        {

            "name": "GET listado de ciudaddes BookingApp",

            "request": {

                "method": "GET",

                "header": []

            },

            "response": []

        },

        {

            "name": "POST inicio de secion de bookingApp",

            "event": [

                {

                    "listen": "test",

                    "script": {

                        "exec": [

                            "pm.test(\"Status code is 200\", function () {\r",

                            "    pm.response.to.have.status(200);\r",

                            "});\r",

                            "\r",

                            "pm.test(\"Response time is less than 900ms\", function () {\r",

                            "    pm.expect(pm.response.responseTime).to.be.below(900);\r",

                            "});\r",

                            "\r",

                            "pm.test(\"Validar mail de patoalloco13@gmail.com\", function () {\r",

                            "    var jsonData = pm.response.json();\r",

                            "    pm.expect(jsonData.email).to.eql(\"patoalloco13@gmail.com\");\r",

                            "});"

                        ],

                        "type": "text/javascript"

                    }

                }

            ],

            "request": {

                "method": "POST",

                "header": [],

                "body": {

                    "mode": "raw",

                    "raw": "{\r\n    \"email\": \"patoalloco13@gmail.com\",\r\n    \"contrasena\": 123456\r\n}",

                    "options": {

                        "raw": {

                            "language": "json"

                        }

                    }

                },

                "url": {

                    "raw": "http://fe.deitech.online:8001/auth/login",

                    "protocol": "http",

                    "host": [

                        "fe",

                        "deitech",

                        "online"

                    ],

                    "port": "8001",

                    "path": [

                        "auth",

                        "login"

                    ]

                }

            },

            "response": []

        }

    ]

}

{

    "info": {

        "\_postman\_id": "6bd3d9db-6f1c-4d80-aafc-0bd6f14d86f2",

        "name": "Final Testing BookingApp - Patricio Alloco",

        "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

        "\_exporter\_id": "18541597"

    },

    "item": [

        {

            "name": "GET listado de ciudaddes BookingApp",

            "request": {

                "method": "GET",

                "header": []

            },

            "response": []

        },

        {

            "name": "POST inicio de secion de bookingApp",

            "event": [

                {

                    "listen": "test",

                    "script": {

                        "exec": [

                            "pm.test(\"Status code is 200\", function () {\r",

                            "    pm.response.to.have.status(200);\r",

                            "});\r",

                            "\r",

                            "pm.test(\"Response time is less than 900ms\", function () {\r",

                            "    pm.expect(pm.response.responseTime).to.be.below(900);\r",

                            "});\r",

                            "\r",

                            "pm.test(\"Validar mail de patoalloco13@gmail.com\", function () {\r",

                            "    var jsonData = pm.response.json();\r",

                            "    pm.expect(jsonData.email).to.eql(\"patoalloco13@gmail.com\");\r",

                            "});"

                        ],

                        "type": "text/javascript"

                    }

                }

            ],

            "request": {

                "method": "POST",

                "header": [],

                "body": {

                    "mode": "raw",

                    "raw": "{\r\n    \"email\": \"patoalloco13@gmail.com\",\r\n    \"contrasena\": 123456\r\n}",

                    "options": {

                        "raw": {

                            "language": "json"

                        }

                    }

                },

                "url": {

                    "raw": "http://fe.deitech.online:8001/auth/login",

                    "protocol": "http",

                    "host": [

                        "fe",

                        "deitech",

                        "online"

                    ],

                    "port": "8001",

                    "path": [

                        "auth",

                        "login"

                    ]

                }

            },

            "response": []

        }

    ]

}